Акционерное общество «Флекс Софтваре Системс»
127055 Россия, г. Москва, ул. Новолесная 2, офис 3
Тел.: +7 (495) 788 - 03 - 25
e-mail: info@flexsoft.com
www.flexsoft.com

Инструкция по эксплуатации программного обеспечения FXL 3.0

Москва 2025

Содержание

1. Конфигурирование и запуск Rest-API программного обеспечения FXL 3.0 на импортонезависимом технологическом стеке (интерфейс для работы ДБО)

Файл сервиса /opt/rest-api/fxl-rest-api-1.0.0.jar

Работа с «DataGrid» осуществляется по порту 8091.

Настройка выполняется в файле /etc/systemd/system/FXL-rest-api.service

```
Environment="JAVA_HOME=/opt/java/jdk-pro-11.0.25-full"
Environment="PORT=8091"
Environment="URL_IGNITE=localhost:10800/TBG"
Environment="SESSION_USER=USR2"
ExecStart=/opt/java/jdk-pro-11.0.25-full/bin/java -jar ./fxl-rest-api-1.0.0.jar -DIGNITE_ALLOW_DML_INSIDE_TRANSACTION=true
```

Где:
PORT – порт обслуживание запросов
URL_IGNITE – JDBC URL «DataGrid»

Запуск и остановка осуществляется через сервис systemctl - FXL-rest-api.service

Лог работы сервиса Rest-API - /var/log/flexsoft/fxl-rest-api.log

2. Служебные скрипты

Наименование скрипта	Назначение скрипта
/opt/ignite/Ignite-SE-17.0.1-beta1/check_fxl-ignite_ready.sh	Проверка завершения запуска «DataGrid». Используется в ExecStartPre сервисов FXL-front-app.service FXL-rest-api.service FXL-sync-oracle2ydb.service FXL-sync-ydb2oracle.service
/opt/ignite/Ignite-SE-17.0.1-beta1/run_ignite_script_cli.sh	Выполнение sql-скриптов в «DataGrid»
/opt/ydb/yasubd-24.3.13.12/drop_create_topics.sh	Пересоздание топиков YDB для синхронизации и отправки сообщений в Мультикарту
/opt/ydb/yasubd-24.3.13.12/check_topics.sh	Проверка существования необходимых топиков YDB

Скрипты с правами на исполнение от пользователя root

Скрипт check_fxl-ignite_ready.sh

```
#!/bin/bash

export JAVA_HOME=/opt/java/jdk-pro-11.0.22
export IGNITE_HOME=/opt/ignite/Ignite-SE-17.0.1-beta1
export IGNITE_URL=grid-122.gondor.fors:21801
export PATH=$JAVA_HOME/bin:$PATH

cd $IGNITE_HOME

SQLMARKER=FXL_IGNITE_READY
SQL2CHECK="SELECT concat('$SQLMARKER ',value) as FXLIGNITEREADY FROM sys.DS_ATOMICLONGS t WHERE t.name =
'CLUSTER_ENV_READY_FLAG_NAME'"
[ ! -z "$1" ] && [ "$1" = "create_indexes" ] && do_create_indexes=yes || do_create_indexes=no
steps=40
step=0
delayonattp=10

while [ $step -le $steps ]; do
   is_ignite_ready=`$IGNITE_HOME/bin/sqlline.sh --outputformat=csv --verbose=true -u "jdbc:ignite:thin://$IGNITE_URL" -n tbg -p tbg -
-fastConnect=true --silent=true \
   -e "$SQL2CHECK" 2> /dev/null | grep "$SQLMARKER" | sed -e "s/'//g" | cut -f 2 -d " "`

   if [ "0$is_ignite_ready" = "01" ]; then
      echo `date +"%Y-%m-%d %H:%M:%S"`" Ignite ready"
      sleep $delayonattp
      break
   else
      echo `date +"%Y-%m-%d %H:%M:%S"`" Ignite not ready"
   fi
   step=`expr $step + 1`
   sleep $delayonattp
done
if [ "0$is_ignite_ready" = "01" ]; then
   echo `date +"%Y-%m-%d %H:%M:%S"`" Ignite successfully started"
   if [ $do_create_indexes = "yes" ]; then
      echo `date +"%Y-%m-%d %H:%M:%S"`" Delay ${delayonattp}sec"
      echo `date +"%Y-%m-%d %H:%M:%S"`" Run create indexes script"
      ./run_ignite_script_cli.sh indexes.sql
      echo `date +"%Y-%m-%d %H:%M:%S"`" Creation indexes script finished"
   fi
   exit 0
```

```
else
   echo `date +"%Y-%m-%d %H:%M:%S"`" Ignite was not started until timeout"
   exit 1
fi
```

Скрипт run_ignite_script_cli.sh. SQL-скрипт передается первым параметром

```
#!/bin/bash

export JAVA_HOME=/opt/java/jdk-pro-11.0.22

/opt/ignite/Ignite-SE-17.0.1-beta1/bin/sqlline.sh --verbose=true -u "jdbc:ignite:thin://127.0.0.1:21801" -n tbg -p tbg --fastConnect=true
--silent=true --run=$1 2> /dev/null
```

Скрипт drop_create_topics.sh

```
#!/bin/bash

ydb_grpc_url=localhost:2136
ydbhome=/opt/ydb/yasubd-24.3.13.12
yqlcmd="$ydbhome/bin/ydb -e -vvv --endpoint grpc://$ydb_grpc_url --database /Root/FXL --user root --no-password "
tables="MGC_SM_DOCUMENTS MGC_TR_ENTRIES SCS_DM_DOCUMENTS"
change_feed_name=CDC_CHANGEFEED0
hours2keep=168

for t in $tables; do
   echo "Drop consumer CDC_${t}_CONSUMER"
   $yqlcmd topic consumer drop --consumer CDC_${t}_CONSUMER TBG/${t}/$change_feed_name
   echo "Drop CHANGEFEED for \`TBG/$t\`"
   $yqlcmd yql -s "ALTER TABLE \`TBG/$t\` DROP CHANGEFEED $change_feed_name"
   echo "Create CHANGEFEED for \`TBG/$t\`"
   $yqlcmd yql -s "ALTER TABLE \`TBG/$t\` ADD CHANGEFEED $change_feed_name WITH ( FORMAT = 'JSON', MODE =
'NEW_AND_OLD_IMAGES', VIRTUAL_TIMESTAMPS = TRUE, RETENTION_PERIOD = Interval('PT${hours2keep}H') )"
   echo "Add consumer CDC_${t}_CONSUMER"
   $yqlcmd topic consumer add --consumer CDC_${t}_CONSUMER TBG/${t}/$change_feed_name
done

tables="INT_TR_ENTRIES"
change_feed_name=CDC_T_INT_TR_ENTRIES
for t in $tables; do
   echo "Drop consumer CDC_${t}_CONSUMER"
   $yqlcmd topic consumer drop --consumer CDC_T_${t}_CONSUMER TBG/${t}/$change_feed_name
   echo "Drop CHANGEFEED for \`TBG/$t\`"
   $yqlcmd yql -s "ALTER TABLE \`TBG/$t\` DROP CHANGEFEED $change_feed_name"
   echo "Create CHANGEFEED for \`TBG/$t\`"
   $yqlcmd yql -s "ALTER TABLE \`TBG/$t\` ADD CHANGEFEED $change_feed_name WITH ( FORMAT = 'JSON', MODE =
'NEW_AND_OLD_IMAGES', VIRTUAL_TIMESTAMPS = TRUE, RETENTION_PERIOD = Interval('PT${hours2keep}H') )"
   echo "Add consumer CDC_${t}_CONSUMER"
   $yqlcmd topic consumer add --consumer CDC_T_${t}_CONSUMER TBG/${t}/$change_feed_name
done
```

Скрипт check_topics.sh

```
#!/bin/bash

ydbhome=/opt/ydb/yasubd-24.3.13.12
export LD_LIBRARY_PATH=$ydbhome/lib
grpc_url=localhost:2136
ydb_database=/Root/FXL
allerrors=0
```

```
checktopic() {
   tbname=$1
   cdcname=$2
   consumer=$3
   echo "Check topic for $tbname"
err=`$ydbhome/bin/ydb -vvv --endpoint grpc://$grpc_url --database $ydb_database --user root --no-password \
   topic read $ydb_database/TBG/$tbname/$cdcname --consumer=$consumer --format=newline-delimited --commit=false --limit 1
2>&1 | grep "The last error was: no path " | grep "Closing session to cluster" | wc -l`
if [ "$err" -ge 1 ]; then
   echo "Not found topic for $tbname"
   allerrors=$((allerrors+1))
else
   echo "Topic for $tbname exists"
fi
}

checktopic INT_TR_ENTRIES CDC_T_INT_TR_ENTRIES CDC_T_INT_TR_ENTRIES_CONSUMER
checktopic MGC_SM_DOCUMENTS CDC_CHANGEFEED0 CDC_T_MGC_SM_DOCUMENTS_CONSUMER
checktopic MGC_TR_ENTRIES CDC_CHANGEFEED0 CDC_T_MGC_TR_ENTRIES_CONSUMER
checktopic SCS_DM_DOCUMENTS CDC_CHANGEFEED0 CDC_T_SCS_DM_DOCUMENTS_CONSUMER

if [ "$allerrors" -gt 0 ]; then
   echo "Not found $allerrors topics"
   exit -1
else
   echo "4 topics are exists"
   exit 0
fi
```

3. Порядок запуска и отслеживания успешности запуска сервисов программного обеспечения FXL 3.0 на импортонезависимом технологическом стеке

3.1. Запуск YDB

YDB storage сервер (сервис ydb-storage.service)

YDB database (сервис ydb-db-FXL.service. Запускать через 5-7 сек после запуска ydb-storage.service)

3.2. Запуск «DataGrid»

(запускать через 5-7 сек после сервисов ydb-storage.service и ydb-db-FXL.service)

3.3. Приложение и REST-сервис программного обеспечения FXL 3.0 на импортонезависимом технологическом стеке

Выполнять запуск ТОЛЬКО после полного запуска «DataGrid» (п.3.2). Критерий полного запуска «DataGrid» это результат запроса из «DataGrid» – *SELECT value FROM sys.DS_ATOMICLONGS t WHERE t.name = 'CLUSTER_ENV_READY_FLAG_NAME'*. Результат должен быть равен «1». Полный запуск «DataGrid» занимает около 3 минут. Контроль завершенности запуска «DataGrid» осуществляется скриптом /opt/ignite/Ignite-SE-17.0.1-beta1/check_fxl-ignite_ready.sh в ExecStartPre в настройках зависимых сервисов

3.4. Отслеживания успешности запуска сервисов программного обеспечения FXL 3.0 на импортонезависимом технологическом стеке

YDB – в логах нет критичных ошибок работы сервиса

«DataGrid» – Запрос п.3.3 возвращает «1» и в логах нет критичных ошибок работы сервиса

Приложение – в логах нет критичных ошибок работы сервиса и наличие в логах информации о запуске:

> … INFO [main] org.apache.catalina.startup.HostConfig.deployWAR **Deployment of web application archive [/opt/axiom-libercat9-9.0.91/webapps/ROOT.war] has finished** in [29 739] ms
>
> … INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8090"]
>
> … INFO [main] org.apache.catalina.startup.Catalina.start **Server startup in [29827] milliseconds**

REST-сервис – Обращение через веб-браузер к <*имя сервера или IP-адрес Rest-API*>:8091/actuator/health/ . В ответе от Rest-API должно быть:

{"status":"UP"}

4. Описание сервисов программного обеспечения FXL 3.0 на импортонезависимом технологическом стеке в systemctl

ydb-storage.service (п.3.1.a)

```
[Unit]
Description=YDB storage service

[Service]
#ExecStartPre=/usr/bin/chown ydb /dev/disk/by-partlabel/ydb_FXL_disk_01
LimitNOFILE=16384
User=ydb
WorkingDirectory=/opt/ydb/yasubd-24.3.13.12-linux-amd64
ExecStart=/opt/ydb/yasubd-24.3.13.12-linux-amd64/bin/ydbd server --log-level 3 --syslog --tcp --yaml-config
/opt/ydb/yasubd-24.3.13.12-linux-amd64/config/config_drive.yaml --grpc-port 2136 --ic-port 19001 --mon-port 8765 --
node static
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=YDB-storage
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target
```

ydb-db-FXL.service (п.3.1.b)

```
[Unit]
Description=YDB FXL database

[Service]
ExecStartPre=/bin/sleep 30
LimitNOFILE=16384
User=ydb
WorkingDirectory=/opt/ydb/yasubd-24.3.13.12-linux-amd64
ExecStart=/opt/ydb/yasubd-24.3.13.12-linux-amd64/bin/ydbd server --yaml-config /opt/ydb/yasubd-24.3.13.12-linux-
amd64/config/config_drive.yaml --tenant /Root/FXL --node-broker vbnkinmdbtst01.bankexp.local:2136 --grpc-port 31011
--ic-port 31013 --mon-port 31012 --log-file-name /opt/ydb/yasubd-24.3.13.12-linux-amd64/logs/FXL_reg.log
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=YDB-FXL-DB
Restart=always
RestartSec=3

[Install]
WantedBy=ydb-storage.service
```

ignite.service (п.3.2)

```
[Unit]
Description=FXL Ignite Service
After=ydb-db-FXL.service
```

```
[Service]
LimitNOFILE=16384
WorkingDirectory=/opt/ignite/Ignite-SE-17.0.1-beta1
User=root
PrivateDevices=yes
ProtectSystem=full
Type=simple
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGTERM
TimeoutStopSec=10
ExecStart=/opt/ignite/Ignite-SE-17.0.1-beta1/start_ignite_service.sh
SyslogIdentifier=FXL-Ignite
Restart=on-failure
RestartSec=5

[Install]
Alias=ignite.service
```

FXL-front-app.service (п.3.3)

```
[Unit]
Description=FXL front application Service
After=ignite.service

[Service]
ExecStartPre=/opt/ignite/Ignite-SE-17.0.1-beta1/check_fxl-ignite_ready.sh
Type=forking
Environment="JAVA_HOME=/opt/java/jdk-pro-11.0.25-full"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djbo.tmpdir=/opt/axiom-libercat9-9.0.91/temp/ -
Djbo.doconnectionpooling=false -Djbo.ampool.initpoolsize=0 -Djbo.ampool.maxpoolsize=100 -
Djbo.ampool.maxavailablesize=100 -Djbo.recyclethreshold=100 -Djbo.ampool.minavailablesize=0 -
Djbo.ampool.timetolive=-1 -Djbo.ampool.maxinactiveage=2400000 -Djbo.ampool.monitorsleepinterval=1200000 -
Djbo.max.cursors=100 -Djbo.locking.mode=pessimistic -Djbo.sql92.LockTrailer= -Djbo.maxpassivationstacksize=100 -
Djbo.txn.disconnect_level=1 -Dcom.flexsoft.fxl.oracleFormsRunMode=JWS -
Dcom.flexsoft.fxl.maxQueryElapsedTime=3000 -Dcom.flexsoft.fxl.useSessionKeepAlive=true -
Dcom.flexsoft.fxl.sessionKeepAliveTimeout=10900000 -
Djbo.def.mgr.listener=com.flexsoft.fxl.masterapp.model.CustomDefMgrListener -Dfxl.dualCoreMode=true -
Doracle.as.jmx.framework.spi=oracle.as.jmx.framework.standardmbeans -
Djbo.SQLBuilder=com.flexsoft.fxl.commons.adf.modelutils.baseclasses.FxlOracleSQLBuilderImpl -
Doracle.mds.cache=simple -Dorg.apache.el.parser.SKIP_IDENTIFIER=true -Djbo.passivationstore=file -
Djbo.tmpdir=/opt/axiom-libercat9-9.0.91/temp/ -Dorg.apache.el.parser.SKIP_IDENTIFIER_CHECK=true "
Environment="CATALINA_PID=/var/run/libercat/%i.pid"
Environment="CATALINA_BASE=/opt/axiom-libercat9-9.0.91/"
Environment="CATALINA_HOME=/opt/axiom-libercat9-9.0.91/"
ExecStart=/opt/axiom-libercat9-9.0.91/bin/startup.sh
ExecStop=/opt/axiom-libercat9-9.0.91/bin/shutdown.sh
WorkingDirectory=/opt/axiom-libercat9-9.0.91
User=root
PrivateDevices=yes
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGTERM
TimeoutStopSec=10
SyslogIdentifier=FXL-front-app
Restart=on-failure
```

```
RestartSec=5

[Install]
Alias=FXL-front-app.service
```

FXL-rest-api.service (п.3.3)

```
[Unit]
[Unit]
Description=FXL rest-API Service
After=ignite.service

[Service]
ExecStartPre=/opt/ignite/Ignite-SE-17.0.1-beta1/check_fxl-ignite_ready.sh
WorkingDirectory=/opt/rest-api
User=root
PrivateDevices=yes
ProtectSystem=full
Type=simple
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGTERM
TimeoutStopSec=10
Environment="JAVA_HOME=/opt/java/jdk-pro-11.0.25-full"
Environment="PORT=8091"
#Environment="PORT=8099"
Environment="URL_IGNITE=localhost:10800/TBG"
Environment="SESSION_USER=USR2"
ExecStart=/opt/java/jdk-pro-11.0.25-full/bin/java -jar ./fxl-rest-api-1.0.0.jar -
DIGNITE_ALLOW_DML_INSIDE_TRANSACTION=true
SyslogIdentifier=FXL-rest-api
Restart=on-failure
RestartSec=5

[Install]
Alias=FXL-rest-api.service
```

5. Управление сервисами производится с помощью утилиты systemctl. Краткая справка по утилите systemctl

start PATTERN...

Запускает (активирует) один или более сервисов (далее «юнитов»), указанных в командной строке. Следует иметь в виду, что шаблоны поиска работают только с первичными именами юнитов, загруженных в память. Юниты, которые неактивны и при этом не завершили работу с ошибкой, обычно не хранятся в памяти, и не будут найдены ни одним шаблоном. Также, в случае экземпляров юнитов, systemd зачастую не знает имена экземпляров, которые еще не были запущены. Следовательно, использование шаблонов поиска со start имеет ограниченное применение. Также не производится поиск по псевдонимам юнитов.

stop PATTERN...

Останавливает (деактивирует) один или более юнитов, указанных в командной строке.

reload PATTERN...

Перезагружает конфигурацию всех юнитов, перечисленных в командной строке. Это затрагивает только конфигурацию связанных служб. Для перезагрузки конфигурации юнита следует использовать команду daemon-reload. Эту команду не следует путать с командой daemon-reload.

restart PATTERN...

Перезапускает один или более юнитов, указанных в командной строке. Если юниты не запущены, они будут запущены.

try-restart PATTERN...

Перезапускает один или более юнитов, указанных в командной строке. Если запущенных юнитов нет, ничего не происходит.

reload-or-restart PATTERN...

Перезагружает один или более юнитов, если они поддерживают данную операцию. В противном случае, перезапускает их. Незапущенные юниты будут запущены.

try-reload-or-restart PATTERN...

Перезагружает один или более юнитов, если они поддерживают данную операцию. В противном случае, перезапускает их. Не затрагивает незапущенные юниты.

isolate NAME

Запускает указанный юнит и все его зависимости, останавливает все остальные юниты. Если имя юнита указано без расширения, предполагается расширение ".target".

Это подобно изменению уровня выполнения в традиционной системе init. Команда isolate немедленно останавливает процессы, которые не включены в новом юните, в том числе, потенциально, используемое графическое окружение или терминал. Следует иметь в виду, что это разрешено только для юнитов с включенным параметром AllowIsolate=. См. systemd.unit(5)

kill PATTERN...

Посылает сигнал одному или более процессам юнита. Для указания конкретных процессов, которые нужно завершить, используется опция --kill-who=. Чтобы указать, какой сигнал отправить, следует использовать --signal=.

is-active PATTERN...

Проверяет, являются ли какие-либо из указанных юнитов активными (например, запущенными). Возвращает код возврата 0, если хотя бы один юнит активен, в ином случае возвращает ненулевое значение. Также выводит информацию о статусе юнитов, если не указана опция --quiet.

is-failed PATTERN...

Проверяет указанные юниты на состояние "failed". Возвращает код возврата 0, если хотя бы один такой юнит найден, в противном случае возвращает ненулевое значение. Также выводит информацию о статусе юнитов, если не указана опция --quiet.

status [PATTERN...|PID...]]

Выводит сжатую информацию о статусе одного или более юнитов, за которой следуют самые недавние данные из журнала. Если юниты не указаны, выводит информацию о состоянии системы. При использовании с опцией --all также выводит статус всех юнитов (с учетом ограничений, наложенных опцией -t). Если в качестве параметра передан PID процесса, будет выведена информация о юните, которому принадлежит данный процесс. Данная функция генерирует человекочитаемый вывод. Для машиночитаемого вывода следует использовать команду show. По умолчанию, единовременно отображается 10 строк вывода, сокращенных многоточиями, чтобы уместиться в окно терминала. Это можно изменить с помощью команд --lines и --full (см. выше). Также стоит заметить, что команды journalctl --unit=NAME или journalctl --user-unit=NAME используют похожий фильтр сообщений и могут быть более удобны в использовании.

show [PATTERN...|JOB...]

Выводит список свойств одного или более юнитов, заданий или самого менеджера. Если не передан аргумент, будут выведены свойства менеджера. Если указано имя юнита, будут выведены свойства данного юнита. Если указан ID задания, будут выведены свойства данного задания. По умолчанию пустые свойства не отображаются, для их отображения следует использовать опцию --all. Для отображения только определенных свойств следует использовать опцию --property=. Команда предназначена для получения машиночитаемого вывода. Для получения человеко читаемого вывода следует использовать команду status.

6. Логирование сервисов программного обеспечения FXL 3.0 на импортонезависимом технологическом стеке

Логи сервера приложений в каталоге /opt/axiom-libercat9-9.0.91/logs

Логи прочих компонентов находятся в каталоге /var/log/flexsoft/

fxl-ignite.log – лог «DataGrid»

fxl-rest-api.log – лог Rest-API

fxl-ydb-storage.log – лог YDB storage server

fxl-ydb-db.log – лог YDB БД

7. Пользовательское взаимодействие с программным обеспечением FXL 3.0

7.1 Форма для работы с файлами входящих и исходящих посылок

Данная форма предназначена для создания исходящих посылок путем отбора подготовленных платежных документов. После создания посылки осуществляется отправка (выгрузка на клиента) готового XML-файла. Для исходящих посылок предусмотрена возможность удаления, в случае если посылка еще не была выгружена на клиента.

Любое содержимое посылок можно посмотреть в форме просмотра реестра документов посылки. Для входящих и исходящих посылок открывается своя форма в зависимости от направления посылки.

Сортировка данных грида осуществляется через группу фильтров, выбирая значения и нажимая кнопку *Поиск* выполняется заполнение грида данными в соответствии с установленными фильтрами.

Внешний вид формы представлен на рис.7.1

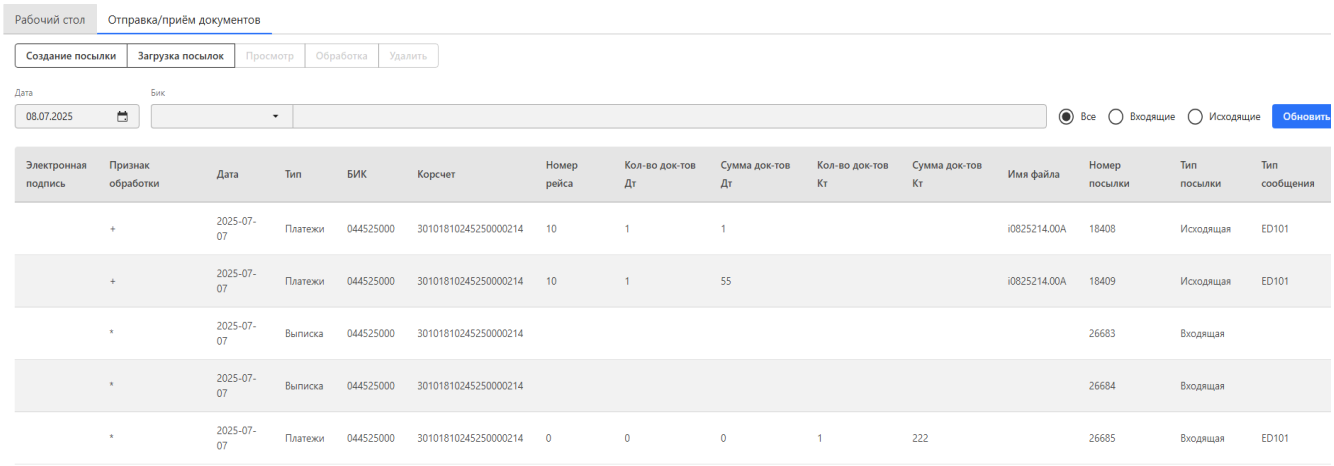


Рис.7.1

Группа функциональных кнопок:

Создание посылки – открытие формы для отбора документов в исходящую посылку

Загрузка посылок – открытие формы для ручной загрузки входящих посылок

Просмотр – открытие формы для просмотра реестра документов в посылке

Обработка – кнопка для создания XML-файла с исходящей посылкой на основе пакета отобранных документов

Удалить – кнопка для удаления посылки из списка

Обновить – кнопка для применения установленных фильтров

Группа фильтров формы:

Дата – поле для отбора посылок за нужную дату. По умолчанию заполнено датой операционного дня пользователя

БИК – поле со списком для отбора посылок по выбранному БИКу банка-корреспондента

Набор радиокнопок «Все, Входящие, Исходящие» – для отбора требуемых типов посылок

Поля списочной формы:

Электронная подпись – признак проставления подписи на посылке

Признак обработки – признак полной обработки посылки. Если посылка полностью обработана, то поле имеет значение «*» - подтвержден; если посылки отправлена, но на неё ещё не получено

подтверждение, то «+» - отправлена; если посылка отправлена и на нее поступила посылка с отрицательным результатом контроля, то «E» - отрицательный результат контроля, иначе – пусто.

Дата – дата загруженной/отправленной посылки (поле EDDate в загружаемом/сформированном XML-файле)

Тип – тип посылки (платежи, подтверждения, выписка, запрос в РКЦ, запрос в банк, ответ банка, справочная информация)

БИК – БИК кредитной организации, отправителя посылки

Корсчет - НОСТРО-счет банка

Номер рейса – номер рейса посылки

Кол-во док-тов Дт – количество дебетовых документов в посылке

Сумма док-тов Дт – сумма дебетовых документов в посылке

Кол-во док-тов Кт – количество кредитовых документов в посылке

Сумма док-тов Кт – сумма кредитовых документов в посылке

Имя файла – поле содержит наименование сформированного файла и тип посылки

Номер посылки - системный номер посылки

Тип посылки – тип посылки: входящая или исходящая

Тип сообщения - тип ED согласно альбому УФЭБС

7.2 Форма создания посылки

Данная форма служит для отбора готовых платежных документов исходя из данных группы направления и группы фильтров. Отбираются платежные документы, содержащие те же реквизиты, которые указаны в фильтре отбора.

Форма вызывается по кнопке «Создать посылку» (рис.7.2).

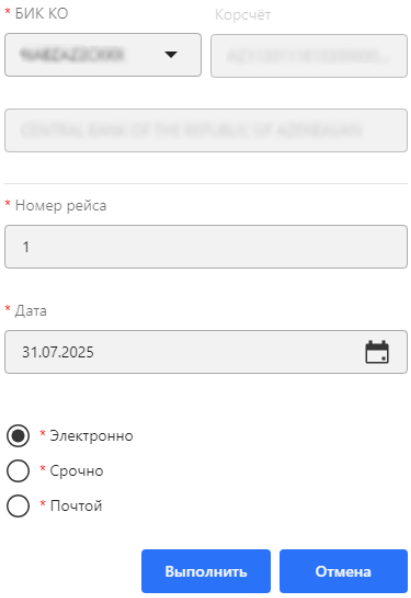


Рис.7.2

В форме заполняются:

- БИК (корсчет и наименование банка заполняются автоматически)
- номер рейса
- дата
- способ перевода

после чего по кнопке *Выполнить* производится отбор документов, подходящих под условия установленных фильтров и создание новой исходящей посылки. После выполнения посылка будет доступна в гриде формы Отправка/прием документов.

По кнопке *Отмена* выполняется выход из формы без выполнения отбора.

7.3 Форма просмотра входящей посылки

Форма просмотра входящих посылок открывается по кнопке «Просмотр» исходя из установленного курсора в гриде на записи входящей посылки.

Данная форма служит для просмотра записей входящих документов.

Форма состоит из мастера – детали - грид с набором полей, деталью является блок *Контроль реквизитов*. Поля детали вычисляются для каждой записи мастер-детали и при переключении курсора на другую запись, автоматически отображает нужный набор данных.

Поля детали являются расчетными и не редактируются. Блок карточка клиента содержит информацию по данным из карточки клиента, блок документ содержит данные из входящего платежа. Значение не совпадающих полей подсвечиваются красным шрифтом.

Внешний вид формы представлен на рис.7.3

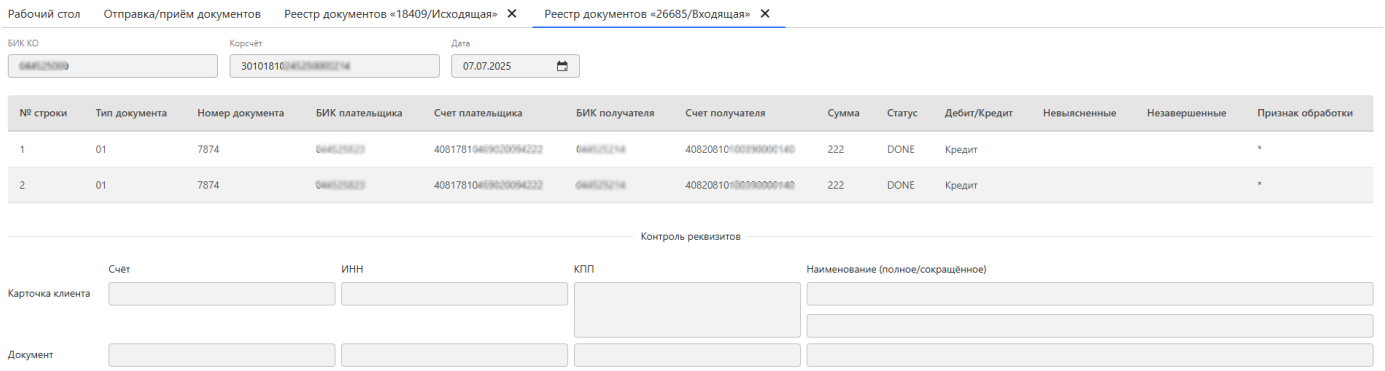


Рис.7.3

Группа справочных полей:

БИК КО – БИК банка корреспондента

Корсчет – корсчет банка корреспондента

Дата – дата посылки

Грид с полями:

№ строки – номер записи по порядку

Тип документа

Номер документа

БИК плательщика

Счет плательщика

БИК получателя

Счет получателя

Сумма

Статус - поле заполняется исходя из статуса основной посылки, а именно если статус основной посылки не заполнен, то статус документа в реестре посылки будет содержать значение *Новая(NEW)*,

если статус основной посылки не пустой, то статус документа в реестре посылки будет содержать значение *Обработано (DONE).*

Дебет/Кредит – признак дебетового/кредитового документа

Невыясненные - признак отнесения документа на счет «до выяснения» (если документ отнесен на счет до выяснения, в поле устанавливается признак *)

Незавершенные - признак отнесения документа на счет незавершенных расчетов (если документ отнесен на счет незавершенных расчетов, в поле устанавливается признак *)

Признак обработки – признак обработки входящего документа. Если документ был авторизован, отнесен на счет «до выяснения» или на счет незавершенных расчетов, то в поле устанавливается признак обработки *.

Деталь к гриду «Контроль реквизитов»:

Группа Карточка клиента

Счет

ИНН

КПП

Наименование (полное/сокращенное)

Группа документ

Счет

ИНН

КПП

Наименование (полное/сокращенное)

7.4 Форма просмотра исходящей посылки

Форма просмотра исходящих посылок открывается по кнопке «Просмотр» исходя из установленного курсора в гриде на записи исходящей посылки. Данная форма служит для просмотра записей исходящих документов.

Внешний вид формы представлен на рис.7.4

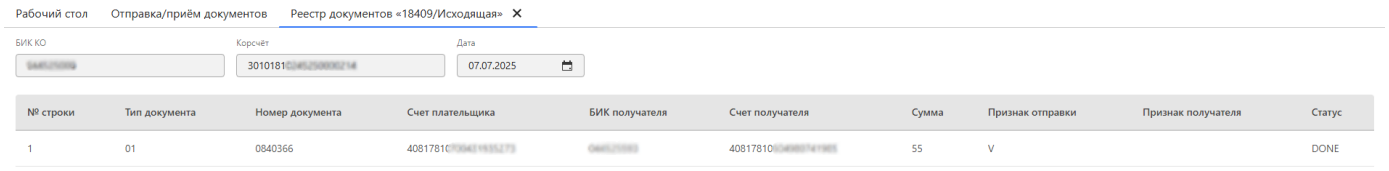


Рис.7.4

Группа справочных полей:

БИК КО – БИК банка корреспондента

Корсчет – корсчет банка корреспондента

Дата – дата посылки

Грид с полями:

№ строки - номер записи по порядку

Тип документа

Номер документа

Счет плательщика

БИК получателя

Счет получателя

Сумма

Признак отправки - поле приобретает статусы в зависимости от статуса самой посылки, если посылка была только создана, но не отправлена (статус основной посылки не заполнен), признак обработки документа посылки не заполняется статусом, если статус основной посылки не пустой, признак обработки документа посылки принимает значение *.

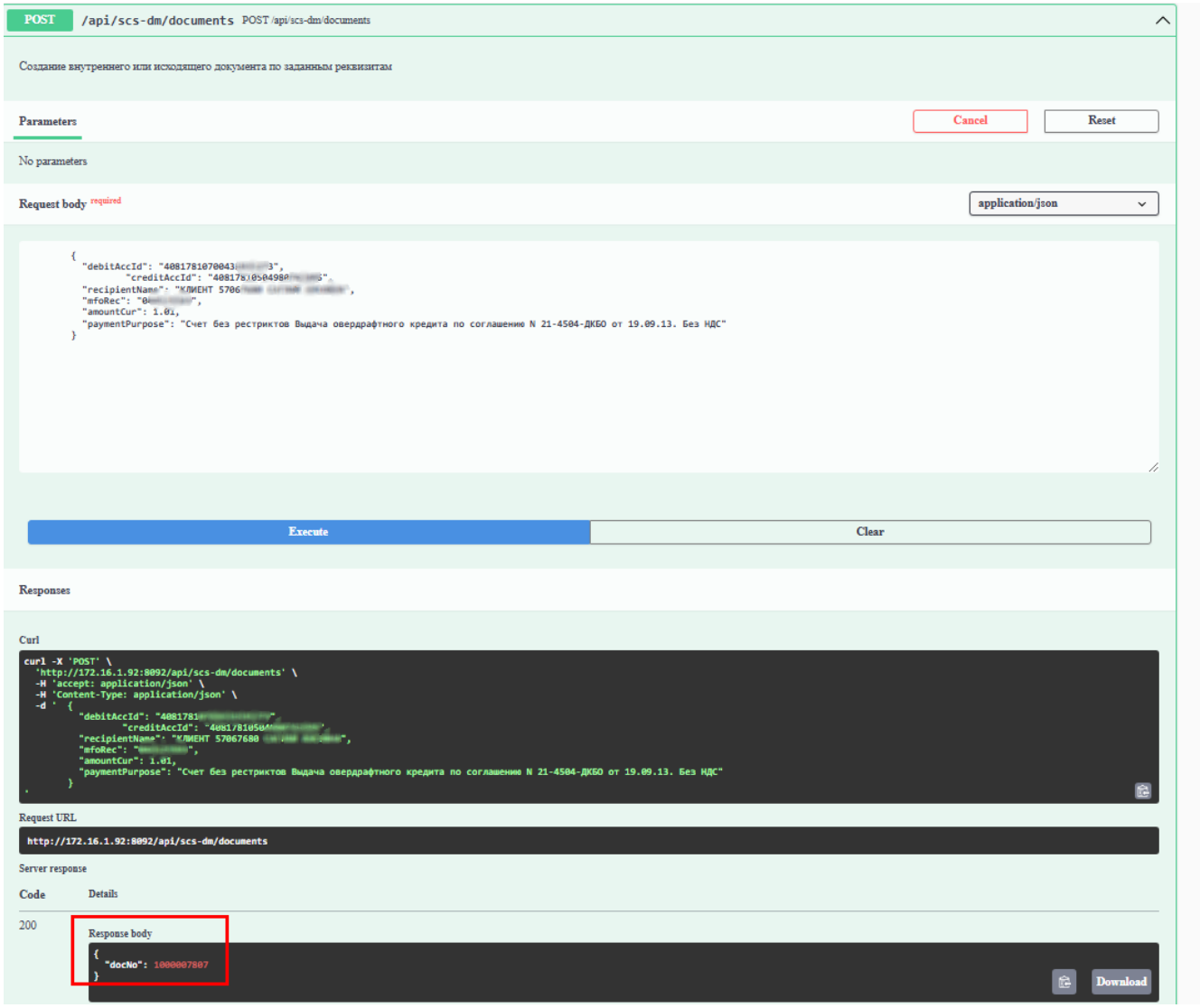
Признак получателя – поле приобретает статус после получения посылок с подтверждением (ED206) либо с отбраковкой платежа (ED201).

Статус – статус платежа (NEW/DONE)

7.5 План тестирования текущего функционала:

Создание исходящего документа:

1. В rest-api путем выполнения метода POST создаем исходящий документ. Дожидаемся вывода системного номера документа в коде 200

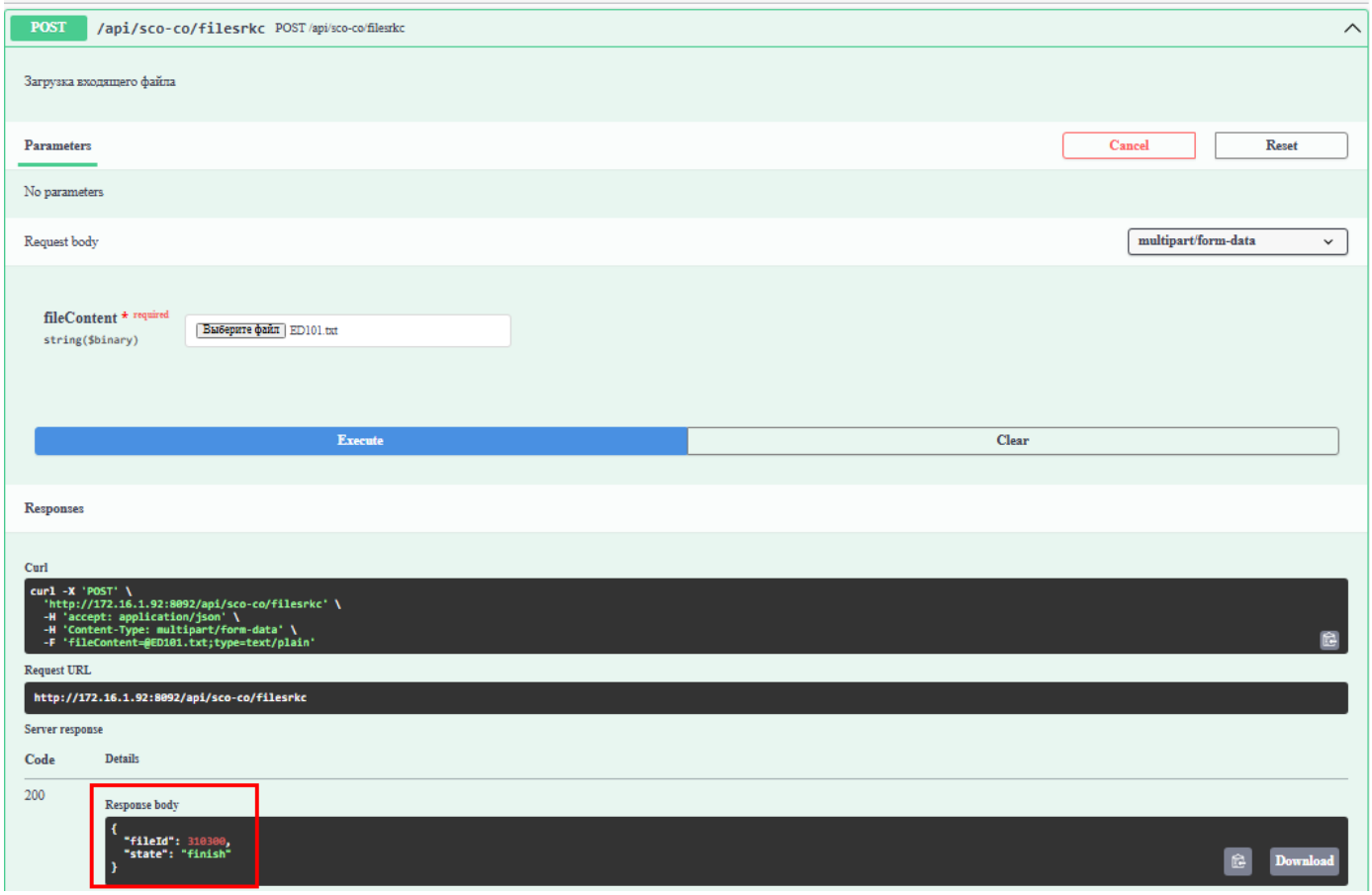


Просмотр исходящей посылки:

1. В FXL открываем форму *Отправка/прием документов*
2. Фильтром **Дата** выбираем дату, в которой есть исходящие посылки - *Выполнить*
3. Выбираем в списке одну из посылок и нажимаем на кнопку *Просмотр,* открывается форма реестра исходящей посылки, проверяем содержимое.

Загрузка входящей посылки:

2. Подготовливаем файл с входящей посылкой в XML-формате.
3. В rest-api путем выполнения метода POST загружаем подготовленный файл, дожидаемся вывода номера посылки в коде 200



4. После загрузки посылка отображается в списочной форме Отправка/прием документов.
5. В FXL открываем форму *Отправка/прием документов*
6. Выбираем в списке загруженную посылку
7. По кнопке *Просмотр* заходим в форму реестра входящей посылки, проверяем содержимое.