УТВЕРЖДЕН
RU.31465715.01000-01 34 01-ЛУ

ПРОГРАММНЫЙ КОМПЛЕКС СЕРВИС ТОЧКА

Сервис «Handoff»

Руководство пользователя

RU.31465715.01000-01 34 01-41

Листов 10

2025

АННОТАЦИЯ

Настоящий документ представляет собой руководство пользователя на Программный комплекс Сервис Точка.

Перечень документов, необходимых для ознакомления, приведен в ведомости эксплуатационных документов «RU.31465715.01000-01 20 01 Программный комплекс Сервис Точка».

В данной части руководства пользователя приведены сведения о назначении и выполнении сервиса «Handoff».

СОДЕРЖАНИЕ

СОКРАЩЕНИЯ, ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

API	Application Programming Interface (программный интерфейс приложения)
ASGI	(Asynchronous Server Gateway Interface) – асинхронный серверный интерфейс для веб-приложения
ESB	(Enterprise Service Bus) – сервисная шина предприятия, архитектурный шаблон интеграции
GUI	Graphical User Interface (графический интерфейс пользователя)
PID	(Process Identifier) – уникальный числовой идентификатор процесса в Linux
SSH	(Secure SHell – защищенная оболочка) – сетевой протокол прикладного уровня, предназначенный для безопасного удаленного доступа к UNIX-системам
БД	База данных
МЧД	Машиночитаемая доверенность
ОС	Операционная система
программа	Сервис «Handoff»
РРФНС	Распределенный реестр федеральной налоговой службы
ФИО	Фамилия, имя, отчество
ФНС	Федеральная налоговая служба

Примечание. Определения, не содержащиеся в настоящем разделе и использующиеся по тексту, имеют значения, установленные для таких определений в документе «Термины и определения», офертах https://tochka.com/offer/ib/ и сети интернет.

1 НАЗНАЧЕНИЕ ПРОГРАММЫ

Программа предназначена для выпуска, подписания МЧД и их регистрации в реестре ФНС России. Функциональные характеристики программы:

- выпуск МЧД;
- подписание МЧД;
- регистрация МЧД в реестре ФНС России;
- поиск МЧД в РРФНС.

2 ВЫПОЛНЕНИЕ ПРОГРАММЫ

2.1 Общие сведения

Условия выполнения приведены в руководстве пользователя «RU.31465715.01000-01 34 01-1 Программный комплекс Сервис Точка. Назначение и условия выполнения Программного комплекса Сервис Точка. Назначение и выполнение сервисов «Авторизационный сервис Точка (crypto-ms)», «Интернет-банк Точка».

Все команды для запуска, выполнения и завершения программы выполняются в консоли ОС Linux.

Linux-консоль представляет собой интерфейс для взаимодействия сотрудника оператора с ОС через текстовые команды. Является мощным инструментом, позволяющим управлять системой, запускать программы, настраивать сервисы и автоматизировать задачи без GUI.

Сотрудник оператора подключается к серверу приложения по протоколу SSH.

2.2 Действия, обеспечивающие загрузку/запуск программы

Последовательность действий, обеспечивающих запуск программы, приведена ниже:

– запуск ESB-шины, см. 2.2.1;
– запуск API-сервера, см. 2.2.2;
– запуск событийной шины, см. 2.2.3.

2.2.1 Запуск ESB-шины

Сотрудник оператора запускает сервис интеграции с ESB-шиной с помощью следующей команды, см. Рисунок 1.

```
python3 -m controllers.esb
```

Рисунок 1 – Команда для запуска ESB-шины

Описание команды:

– запускает сервис интеграции с ESB-шиной – централизованным сервисом

для интеграции передачи данных между сервисами;

– обрабатывает запросы и события.

2.2.2 Запуск API-сервера

Сотрудник оператора запускает API-сервис с помощью следующей команды, см. Рисунок 2.

```
uvicorn controllers.web:app
```

Рисунок 2 – Команда для запуска API-сервера

Описание команды:

– запускает серверный интерфейс ASGI для эффективной асинхронной работы API-сервера;

– путь к приложению: модуль controllers.web, переменная app (экземпляр FastAPI).

2.2.3 Запуск событийной шины

Сотрудник оператора запускает сервис – внутренний обработчик сообщений – с помощью следующей команды, см. Рисунок 3.

```
python3 -m controllers.amqp listen-internal-task-queues
```

Рисунок 3 – Команда для запуска событийной шины

Описание команды:

– запускает обработчик очередей сообщений брокера RabbitMQ;

– слушает сообщения.

2.3 Действия, обеспечивающие выполнение программы

Последовательность действий, обеспечивающих выполнение программы, приведена ниже.

Для управления выполнением программы сотруднику оператора доступны консольные команды, перечень которых приведен в таблице, см. Таблица 1.

Таблица 1 – Перечень, наименование и описание команд для работы с программой

№	Наименование команды	Описание команды
1	ca-sign-transaction	С помощью команды осуществляется подпись транзакции через сервис «Ca-gateway».
2	checking-full-name-for-changes	С помощью команды осуществляется сверка ФИО в МЧД и в сервисе «ApiBank». Можно получить отчетом или отозвать.
3	load-powers	С помощью команды осуществляется загрузка полномочий в БД.
4	revoke-poa-fio	С помощью команды осуществляется принудительный отзыв МЧД из-за смены ФИО.
5	schedule-sync	С помощью команды осуществляется синхронизация статусов МЧД с РРФНС.
6	schedule-upload	С помощью команды осуществляется синхронизация МЧД с РРФНС.
7	set-state	С помощью команды осуществляется смена статуса МЧД с необходимыми действиями и событиями.

2.3.1 Запуск команды управления

Для смены вручную статуса МЧД предусмотрена команда set-state в контроллере amqp, см. Рисунок 4.

```
python3 -m controllers.amqp set-state POA_ID STATE
```

Рисунок 4 – Команда для смены статуса МЧД

Пример вызова команды для смены статуса МЧД приведен ниже, см. Рисунок 5.

```
python3 -m controllers.amqp set-state 1337 revoked
```

Рисунок 5 – Пример вызова команды управления

2.3.2 Ответы программы на команды управления

Пример ответа программы на команду подписи транзакции через сервис «Ca-gateway» приведен ниже, см. Рисунок 6.

```
<event
        refer="ca-gateway-e6[redacted]"
        sender="ca-gateway"
        type="caSignTransaction"
        timestamp="2022-02-11T10:19:51.598+05:00"
        extkey="65[redacted]"
>
    <caSignTransaction
            <!--Идентификатор транзакции-->
            transactionId="22[redacted]"
            <!--Колвир код клиента-->
            customerCode="30[redacted]"
            <!--
                Результат исполнения
                - success - успешно
                - failed - неудачно
                - hidden - транзакция скрыта через админку
                - in_progress - отмена скрытия через админку
            -->
            result="failed"

            <!--
                Опциональная причина, если result="failed"
                - declined - пользователь явно отклонил транзакцию
                - expired - истек таймаут ожидания подтверждения транзакции
                - unknown - другое
            -->
            reason="declined"
            <!--
                Опциональный флаг обозначающий нахождение транзакции в архиве,
                если не установлен, значит false (не в архиве)
            -->
            archived="true"
    />
</event>
```

Рисунок 6 – Пример ответа программы

2.4 Действия, обеспечивающие завершение программы

Последовательность действий, обеспечивающих завершение программы,

приведена ниже:

– поиск процесса, см. 2.4.1;

– убийство процесса, см. 2.4.2.

2.4.1 Поиск процесса

Сотрудник оператора осуществляет поиск подходящего процесса с помощью одной из следующих команд, см. Рисунок 7.

```
ps aux | grep "название"  # Поиск по имени
pgrep -f "название"       # Быстрый поиск PID
pidof программа           # Найти PID программы (например, `pidof python`)
```

Рисунок 7 – Пример команд для поиска процесса

2.4.2 Убийство процесса

Сотрудник оператора осуществляет убийство нужного процесса с помощью одной из следующих команд, приведенных ниже:

– обычное корректное завершение процесса осуществляется с помощью следующей команды, см. Рисунок 8;

```
kill PID  # или kill -15 PID
```

Рисунок 8 – Пример команды для убийства процесса

– принудительное завершение процесса при его зависании осуществляется с помощью следующей команды, см. Рисунок 9;

```
kill -9 PID
```

Рисунок 9 – Пример команды для убийства процесса

– завершение процесса по его имени осуществляется с помощью следующей команды, см. Рисунок 10.

```
pkill -f "название"  # Пример: pkill -f "python script.py"
killall программа    # Пример: killall python
```

Рисунок 10 – Пример команды для убийства процесса