

СВЯЗЬКОМ.ОТА

Платформа для управления данными на SIM-карте

Функциональные характеристики программного обеспечения

Оглавление

Оглавление	2
1 Общие сведения	3
1.1 Функциональные возможности	3
1.2 Функциональные возможности по управлению данными на (e/U)SIM	4
2 Требования к оборудованию	5
3 Интеграция с узлами сигнальной сети	5
4 Описание HTTP API	6
4.1 Функциональность API	6
4.2 Общая информация	6
4.3 Типы операций для входящих API запросов	7
4.3.1 Операция «get_info»	7
4.3.2 Операция «task_exec»	8
4.3.3 Операция «update_common_data»	10
4.3.4 Операция «db_tm_track_show»	10
4.3.5 Операция «ldb_request»	10
4.3.6 Операция «counters_g_get»	11
4.3.7 Операция «counters_g_clear»	11
4.3.8 Операция «campaign_oper»	11
4.3.9 Операция «task_stop»	12
4.4 Типы операций для исходящих API-запросов	12
4.4.1 Операция «envelope_callback_before» и «envelope_callback_after»	12
4.4.2 Операция «task_callback_before»	12
4.4.3 Операция «task_callback_after»	13
4.4.4 Тип операции «get_info_by».	14

1 Общие сведения

1.1 Функциональные возможности

Платформа «Связьком.ОТА» выполняет базовую реализацию ОТА-функций, а именно:

- Реализация механизма безопасности ОТА в соответствии с ETSI TS 123 048 в части SMS-транспорта.
- Поддержка HTTPS в соответствии со спецификацией «Remote Application Management over HTTP Card Specification v2.2».
- Обеспечение защищенного канала связи с приложениями на (e/U)SIM-картах. В том числе с такими приложениями как:
 - RFM (SIM/USIM/ISIM file system);
 - RAM (Remote Application Management);
 - Java Card апплетами;
 - любыми иными приложениями, установленными на (e/U)SIM.
- Выполнение и контроль ОТА Envelope на (U)SIM, получение результата его выполнения.
- Выполнение и контроль ОТА заданий (последовательные ОТА Envelope операции).
- Выполнение и контроль ОТА кампаний (массовое выполнение ОТА-заданий).
- Обработка PoR SMS от (e/U)SIM.
- Использование SMPP v3.4 для взаимодействия с SMSC в части отправки и получения sms-сообщений.
- Детализация и статистическая отчетность.
- Режим постоянного проведения ОТА кампаний, отправка ОТА заданий по мере появления абонентов в сети.
- Реализация HTTP API:
 - Получения данных о внутренних состояниях модуля:
 - состояния соединений к базе данных;
 - состояние SMPP соединения к SMSC;
 - размеры внутренних очередей;
 - прочее.
 - Для запуска ОТА-заданий.
 - HTTP callback запросы на:
 - завершение ОТА задания (с кодом его завершения);
 - начало и окончание ОТА посылки;
 - Управление ОТА-кампаниями:
 - запуск;
 - приостановка;
 - возобновление;
 - Получение значений статистических счетчиков.

1.2 Функциональные возможности по управлению данными на (e/U)SIM

- Чтение и управление файловой системой. Например, реализация следующих задач (RFM):
 - Смена IMSI.
 - Смена SPN (имя сети оператора).
 - Обновление различных PLMN списков, в том числе списка приоритетных сетей для регистрации в роуминге.
 - Обновление номера SMSC.
 - Обновление записной книжки (e/U)SIM карты.
- Управление приложениями, включает в себя (RAM):
 - Получение свободного места в RAM и EEPROM на карте.
 - Получение списка установленных пакетов и приложений.
 - Установка и удаление Java Card апплетов.
- Коммуникация с установленными Java Card апплетами и приложениями. к примеру реализация следующих задач:
 - Реализация услуги «Живой баланс».
 - Реализация DSTK меню.
 - Перерегистрация мобильного телефона в сети.
 - Получения данных о состоянии регистрации аппарата в сети (включая данные о базовой станции, уровнях сигнала и времени распространения сигнала до базовой станции).
 - Выполнение Proactive операций на телефоне и получения ответов на них.

2 Требования к оборудованию

Платформа имеет модульную архитектуру, что обеспечивает её отличную масштабируемость. Аппаратное обеспечение платформы «Связьком.ОТА» выбирается исходя из планируемой нагрузки, а также из требований по сроку хранения детальной информации о работе платформы. Основными лицензируемыми параметрами являются:

- 1) максимальная интенсивность отправки ОТА-команд;
- 2) срок хранения логов;
- 3) срок хранения детализации в БД.

Для повышения надёжности платформа «Связьком.ОТА» устанавливается на кластер. Производительность серверов рассчитывается, исходя из того, что каждый из них может выполнять все функции платформы.

Пример производительности сервера ОТА для следующих требований:

- Производительность, TPS: 100.
- Срок хранения логов, дней: 14.
- Срок хранения данных о транзакциях ОТА, дней: 180.

Спецификация:

- CPU – 4 ядра.
- RAM – 16 Гб.
- Storage – 0,5 Тб (600 IOPS).

3 Интеграция с узлами сигнальной сети

Для обмена SMS платформа интегрируется с сетью оператора по протоколу SMPP. При необходимости платформа может быть дополнена собственным SMSC в составе платформы («Связьком.SMSC»), который интегрируется с сигнальной сетью оператора (STP) по протоколам MAP/M3UA/SCTP и самостоятельно доставляет ОТА SMS абонентам.

Для интеграции с другими узлами в целях реализации основных услуг используется HTTP API.

4 Описание HTTP API

4.1 Функциональность API

Платформа предоставляет входящий и исходящий API.

Входящий API:

- Запуск OTA-заданий и получение результата их выполнения.
- OTA-кампании:
 - Запуск OTA-заданий.
 - Запуск OTA-кампаний.
 - Приостановка OTA-кампаний.
 - Получение информации о состоянии запущенных OTA-кампаний.
- Получение внутренних статусов состояния основного процесса.
 - Статус SMPP-подключения.
 - Размеры различных очередей.
 - Состояния активных HTTP-соединений.
- Получение значений «Переменных», которые используются в OTA-заданиях, если они определены.
- Получение статистических метрик (счетчиков).
- Общее управление OTA-кампаниями и другими настройками платформы.

Исходящий API:

- Промежуточные HTTP callback запросы в процессе выполнения OTA-задания.
- Получение связок ICCID-MSISDN, IMSI-MSISDN из внешней подсистемы.

4.2 Общая информация

- API поддерживает как входящие HTTP-запросы, так и производит HTTP-запросы на внешнюю подсистему для получения или передачи ей какой-либо информации.
- Тело HTTP-запроса и HTTP-ответ являются JSON структурами. Соответственно, во всех запросах должен присутствовать заголовок «Content-type: application/json».
- API поддерживает несколько типов операций для входящих запросов, название операции должно присутствовать в поле «operation». К примеру, операция «get_info»:

```
{
  'operation': "get_info"
}
```

- Результат каждой операции сопровождается обязательной структурой:

```
{
  "result_code": <Код результата>,
  "result_text": "<Описание кода результата>"
}
```

- Код результата и описание кода результата зависят от типа операции.

- Каждый тип операции может нести обязательные или опциональные параметры, которые должны определяться переменными на том же JSON уровне, что и тип операции.
- Результат каждой операции может нести обязательные и опциональные параметры, которые будут определены на том же JSON уровне, что и результат выполнения операции.
- TCP порт для обработки входящих API запросов задается в конфигурационном файле в параметре `http_api_port`. По-умолчанию - 8000

4.3 Типы операций для входящих API запросов

4.3.1 Операция «get_info»

Операция предназначена для получения внутреннего состояния модуля. Возвращает структурированный параметр «**info**» с запрашиваемыми данными.

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
<code>task_sessions</code>	Запросить весь список выполняемых задач	Boolean Integer	Опциональный
<code>env_sessions_ids</code>	Запросить состояние внутренней структуры env_sessions_ids (для целей отладки)	Boolean Integer	Опциональный
<code>env_sessions_sm_id</code>	Запросить состояние внутренней структуры env_sessions_sm_id (для целей отладки)	Boolean Integer	Опциональный
<code>camp_sessions</code>	Запросить состояние внутренней структуры camp_sessions – состояние выполнения ОТА-кампаний	Boolean Integer	Опциональный
<code>ldb</code>	Запросить внутреннее состояние подмодуля LDB	Boolean Integer	Опциональный

Параметры структуры «info»:

Название параметра	Описание параметра и типы принимаемых значений	Тип
<code>db_connection_status</code>	Состояние соединения с БД	Boolean
<code>env_sessions_ids_len</code>	Размер внутренней структуры env_sessions_ids (для целей отладки)	Integer
<code>env_sessions_sm_id_len</code>	Размер внутренней структуры env_sessions_sm_id (для целей отладки)	Integer
<code>http_connections_len</code>	Количество открытых HTTP-соединений, как входящих, так и исходящих	Integer
<code>sm_service.recipient_sm_len</code>	Размер внутренней структуры sm_service.recipient_sm (для целей отладки)	Integer
<code>sm_service.smpp.state</code>	Код состояния SMPP-соединения.	Integer

sm_service.smpp.state_desc	Описание состояния SMPP- соединения.	String
sm_service.submit_sm_seq_len	Длина очереди ожидания команды submit_sm_resp по SMPP	Integer
sm_service.tx_queue_len	Длина очереди SMS на отправку	Integer
task_sessions_ids_len	Количество выполняемых OTA-заданий	Integer
task_sessions_int_ids_len	Размер внутренней структуры task_sessions_int_ids (для целей отладки)	Integer
camp_sessions	Структура camp_sessions , описывающая состояние OTA-кампании	Struct
ldb_init	Состояние инициализации модуля LDB	Boolean
state	Перечень значимых внутренних параметров.	Struct

Пример выполнения операции:

```
{
  "info": {
    "db_connection_status": true,
    "env_sessions_ids_len": 0,
    "env_sessions_sm_id_len": 0,
    "http_connections_len": 1,
    "sm_service.recipient_sm_len": 0,
    "sm_service.smpp.state": 4,
    "sm_service.smpp.state_desc": "BOUND_TRX",
    "sm_service.submit_sm_seq_len": 0,
    "sm_service.tx_queue_len": 0,
    "task_sessions_ids_len": 0,
    "task_sessions_int_ids_len": 0
  },
  "result_code": 0,
  "result_text": "Success"
}
```

4.3.2 Операция «task_exec»

Операция предназначена для запуска заранее подготовленных OTA-заданий.

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
task_id	Числовой идентификатор OTA-задания	Integer	Опциональный
task_name	Название OTA-задания	String	Опциональный
msisdn	MSISDN абонента	String	Опциональный
imsi	IMSI абонента	Integer	Опциональный
iccid	ICCID (U)SIM-карты абонента	Integer	Опциональный
callback_before	Выполнить HTTP-запрос по указанному URL перед выполнением задания	String	Опциональный
callback_after	Выполнить HTTP-запрос по указанному URL после выполнения OTA задания. Запрос будет содержать детальную	String	Опциональный

	информацию о результате выполнения OTA задания.		
step	Указать номер шага, с которого необходимо будет выполнять OTA задание	Integer	Опциональный
vars	Указать значение переменных, которые будут подставляться в OTA Envelope, если они там определены. Пример: {'EF_IMSI':'6F07'}	Struct	Опциональный
card_select_mode	Выбрать режим выбора карт (см. «Режимы выбора карт»)	Integer	Опциональный
user_data	Произвольная структура данных, которая будет добавляться во все исходящие HTTP API запросы в рамках выполнения этого OTA-задания	Struct	Опциональный
src_id	Числовой идентификатор источника. Перечень источников задается в WEB-интерфейсе. По-умолчанию: «WEB»	—	—
channel_id	Идентификатор транспорта. 1 – SMS, 2 – HTTPS	Integer	Опциональный
smpp_validity_period_sec	Устанавливает validity_period в секундах для отправки SMS по SMPP	Integer	Опциональный
sms_por_timeout	Максимальное время ожидания PoR SMS после отправки OTA-посылки	Integer	Опциональный
stop_tasks_before	Остановить запущенные задания для выбранной ICCID карты.	Boolean	Опциональный
application_id	Выбрать ID приложения на которое будут отправлены все посылки этого задания.	Integer	Опциональный

Особенности:

- Один из параметров task_id или task_id является обязательным.
- Один из параметров msisdn, imsi, или iccid является обязательным.

Пример ответа API после успешного запуска задания:

```
{
  "reg_task_id": 149,
  "result_code": 0,
  "result_text": "Task start successful"
}
```

Пример ответа API после неуспешного запуска задания:

```
{
  "result_code": 130,
  "result_text": "SIM card not found"
}
```

4.3.3 Операция «update_common_data»

Операция предназначена для оперативного обновления кэша служебных данных из БД. Во время операции происходит обновление следующих данных:

- Список и параметры приложений
- Список и параметры профилей
- Связки профилей и приложений
- Список используемых ОТА-посылок в ОТА-заданиях
- Условия применения ОТА-посылок для приложений
- Условия применения ОТА-посылок для профилей
- Список и параметры ОТА-заданий
- Список номеров отправителей для SMS
- Список RegEx выражений
- Список и параметры транспортных ключей
- Список источников
- Список и описание кодов завершения задания (status_id)

4.3.4 Операция «db_tm_track_show»

Операция предназначена для вывода в LOG информации о времени выполнения SQL-запросов в БД. Функция предназначена для отладки в условиях высокой нагрузки.

4.3.5 Операция «ldb_request»

Операция предназначена для получения данных из LDB.

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный/опциональный
subscriber_id	MSISDN или IMSI абонента	String	Обязательный

Выходные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный/опциональный
msisdn	MSISDN абонента	String	Обязательный
imsi	IMSI абонента	String	Обязательный
RN	RN абонента	String	Обязательный
billing_type	Тип биллинговой системы абонента	String	Обязательный

4.3.6 Операция «counters_g_get»

Операция предназначена для получения глобальных статистических накопительных данных.

Выходные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный/опциональный
counters_g	Структура с данными статистики	Struct	Обязательный

Пример выполнения операции:

```
{
  "counters_g": {
    "api": 5,
    "api_task_exec": 3,
    "db_sql_exec": 1282,
    "db_sql_exec_ok": 1282,
    "smpp_deliver_sm": 4,
    "smpp_delivery_report": 4,
    "smpp_delivery_report_ok": 2,
    "smpp_por": 2,
    "smpp_submit_sm": 2,
    "smpp_submit_sm_resp": 2,
    "smpp_submit_sm_resp_ok": 2,
    "start": 1,
    "task_exec": 3,
    "task_finished": 2,
    "task_finished_ok": 2
  },
  "result_code": 0,
  "result_text": "Success"
}
```

4.3.7 Операция «counters_g_clear»

Операция предназначена для очистки глобальных статистических накопительных данных.

4.3.8 Операция «campaign_oper»

Операция предназначена для выполнения действий с OTA-кампаниями.

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
campaign_name	Название OTA-кампании	String	опциональный
campaign_id	Числовой идентификатор OTA-кампании	Integer	опциональный
oper	Операция	String	обязательный

Доступные типы операций с OTA кампаниями:

Название	Описание
start	Запустить OTA-кампанию
stop	Остановить OTA-кампанию
suspend	Приостановить OTA-кампанию
resume	Возобновить приостановленную OTA-кампанию

4.3.9 Операция «task_stop»

Операция предназначена для остановки запущенных заданий для выбранного ICCID

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
iccid	ICCID карты абонента	Integer	Обязательный
reg_task_id	reg_task_id конкретного задания. Если reg_task_id не указан, то будут остановлены все задания для выбранного ICCID	Integer	Опциональный

Выходные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
tasks_stoped	Количество остановленных заданий	Integer	Обязательный

Пример ответа API после успешного запуска задания:

```
{
  " tasks_stoped": 1,
  "result_code": 0,
  "result_text": "Successful"
}
```

4.4 Типы операций для исходящих API-запросов

4.4.1 Операция «envelope_callback_before» и «envelope_callback_after»

Операция предназначена для уведомления внешней системы о начале и окончании выполнения OTA-посылки в рамках OTA-задания.

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
callback_name	Название операции	—	—
step	Номер шага, соответствующий OTA посылке	Integer	Обязательный
msisdn	MSISDN абонента	String	Обязательный
imsi	IMSI абонента	String	Обязательный
iccid	ICCID карты абонента	String	Обязательный
profile_id	profile_id профиля карты	Integer	Обязательный

4.4.2 Операция «task_callback_before»

Операция предназначена для уведомления внешней системы о начале выполнения OTA задания.

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
callback_name	Название операции	–	–
reg_task_id	Числовой идентификатор в журнале OTA заданий.	Integer	Обязательный
msisdn	MSISDN абонента	String	Обязательный
imsi	IMSI абонента	String	Обязательный
iccid	ICCID карты абонента	String	Обязательный
profile_id	profile_id профиля карты	Integer	Обязательный
user_data	Произвольная структура данных, которая была передана при выполнении операции «task_exec»	Struct	Опциональный
campaign_id	ID OTA кампании, если OTA задание было выполнено в рамках OTA кампании.	Integer	Опциональный

4.4.3 Операция «task_callback_after»

Операция предназначена для передачи детальной информации о результате выполнения OTA-задания.

Входные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
callback_name	Название операции	–	–
reg_task_id	Числовой идентификатор в журнале OTA заданий	Integer	Обязательный
msisdn	MSISDN абонента	String	Обязательный
imsi	IMSI абонента	String	Обязательный
iccid	ICCID карты абонента	String	Обязательный
steps_processed	Количество шагов (посылок), успешно выполненных в рамках OTA-задания	Integer	Обязательный
status_id	Код результата выполнения OTA-задания	Integer	Обязательный
user_data	Произвольная структура данных, которая была передана при выполнении операции «task_exec»	Struct	Опциональный
envelope_results	Структура с детальным результатом выполнения каждой OTA-посылки	Struct	Обязательный
profile_id	profile_id профиля карты	Integer	Обязательный
campaign_id	ID OTA кампании, если OTA задание было выполнено в рамках OTA кампании.	Integer	Опциональный

Пример выполнения операции после успешного выполнения ОТА-задания:

```
{
  "callback_name": "task_callback_after",
  "envelope_results": [
    {
      "apdu_num": 3,
      "apdu_response": "9000",
      "cntr": 362,
      "data": "xxxxxxxxxxxxxxxxxxxx",
      "reg_envelope_id": 146,
      "status_code": "00",
      "step": 0,
      "tar": "B00010"
    }
  ],
  "iccid": "xxxxxxxxxxxxxxxxxxxx",
  "imsi": "yyyyyyyyyyyyyyyy",
  "msisdn": "zzzzzzzzzz",
  "reg_task_id": 149,
  "status_id": 1,
  "steps_processed": 1,
  "user_data": null
}
```

4.4.4 Тип операции «get_info_by».

Операция предназначена для получения связок MSISDN/IMSI/ICCID от внешней платформы. Доступны следующие операции данного типа:

- get_info_by_msisdn
- get_info_by_imsi
- get_info_by_iccid

URL для выполнения операций задается в конфигурационном файле в параметре **get_info_url**.

Результатом выполнения операции на внешней системе должна быть JSON структура с стандартным результатом выполнения операции и выходными параметрами на том же JSON уровне.

Входные параметры:

Название операции	Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный / опциональный
-	callback_name	Название операции	String	Обязательный
get_info_by_msisdn	msisdn	MSISDN абонента	String Integer	Опциональный
get_info_by_imsi	imsi	IMSI абонента	String Integer	Опциональный
get_info_by_iccid	iccid	ICCID абонента	String Integer	Опциональный

Выходные параметры:

Название параметра	Описание параметра и типы принимаемых значений	Тип	Обязательный/опциональный
msisdn	MSISDN абонента	String	Опциональный
imsi	IMSI абонента	String	Опциональный
iccid	ICCID абонента	String	Опциональный

Таблица обязательности выходных параметров в зависимости от названия операции и режима выбора карт.

Название операции	Название параметра	Режим выбора карт	Обязательный / опциональный
get_info_by_msisdn	imsi	CARD_SELECT_MODE2	Обязательный
get_info_by_msisdn	iccid	CARD_SELECT_MODE2	Опциональный
get_info_by_msisdn	imsi	CARD_SELECT_MODE3	Опциональный
get_info_by_msisdn	iccid	CARD_SELECT_MODE3	Обязательный
get_info_by_imsi	msisdn	CARD_SELECT_MODE2	Обязательный
get_info_by_imsi	iccid	CARD_SELECT_MODE2	Опциональный
get_info_by_imsi	msisdn	CARD_SELECT_MODE3	Обязательный
get_info_by_imsi	iccid	CARD_SELECT_MODE3	Обязательный
get_info_by_iccid	msisdn	CARD_SELECT_MODE3	Обязательный
get_info_by_iccid	imsi	CARD_SELECT_MODE3	Опциональный