

Сервисный режим

- [Агент](#)
- [Сервис](#)
- [Сравнение хэш-сумм метрик за период \(СТАРЫЙ НЕ ИСПОЛЬЗОВАТЬ\)](#)
- [API 1C](#)
- [Сравнение хэш-сумм метрик за период](#)

Агент

Переменные среды

CFG_AGENT_SERVICE_MODE

- 0 - флаг выключенного сервисного режима. **значение по умолчанию**
- 1 - флаг включенного сервисного режима

CFG_AGENT_SERVICE_MODE_URL_BASE

url для получения команд в сервисном режиме

CFG_AGENT_SERVICE_MODE_URL_DATA

url для дополнительной выгрузки (в сервис) метрик в сервисном режиме

CFG_AGENT_SERVICE_MODE_TOKEN

токен сервиса в сервисном режиме

CFG_AGENT_SERVICE_MODE_SYNC_METRIC_CRON

крон расписание для синхронизации метрик коробки и сервиса. Значение по умолчанию - "**10 * * * ***"

CFG_AGENT_SERVICE_MODE_METRIC_SYNC_PERIOD

количество дней для синхронизации метрик коробки и сервиса по расписанию. Значение по умолчанию - **14**

CFG_AGENT_SERVICE_MODE_METRIC_SYNC_CHUNK_MIN

минимальное количество дней в запросе сумм и метрик при синхронизации. Значение по умолчанию - **1**

CFG_AGENT_SERVICE_MODE_METRIC_SYNC_CHUNK_MAX

максимальное количество дней в запросе сумм и метрик при синхронизации. Значение по умолчанию - **7**

Команды

service_mode.sync_metrics

команда синхронизации с произвольным промежутком

Параметры:

- **from** - начало промежутка. utc unixtime
- **to** - окончание промежутка. utc unixtime
- **force** - флаг принудительной синхронизации. при force == True поиск различий не производится и выгружаются все метрики

service_mode.sync_object_struct

команда для передачи измененных объектов из сервиса в коробку. команда-singleton - агент не добавит команду, если в очереди команд есть команда с таким типом.

Параметры

- **object_struct** - строка с бинарными данными структуры объекта в формате base64

API сервиса

GET: /agent/service_mode/get_sum

[wiki](#)

запрос сумм метрик за промежуток времени

- **from** - начало промежутка. utc unixtime
- **to** - окончание промежутка. utc unixtime

GET: /agent/service_mode/get_metrics

[wiki](#)

запрос метрик за промежуток времени

- **from** - начало промежутка. utc unixtime
- **to** - окончание промежутка. utc unixtime

API 1с

POST: /agent/service_mode/object_struct

Отправка измененных данных объектов из сервиса в коробочное издание.

```
{
  "object_struct": "BASE64 строка"
}
```

POST: /agent/service_mode/metrics

Отправка метрик из сервиса в коробку. Предполагается пересылка без изменения формата

Пример тела запроса

```
{
  "client_id": "de61e105- f743- 44aa- b1ab- f799cd513229",
  "db_name": "DB_2",
  "from": 1679443000,
  "to": 1679529000,
  "metrics": [
    ...,
    {
      "times": "2022- 01- 31T11: 00: 00",
      "device_id": "a26c126b- 7372- 4123- a825- 55d1526d1b1d",
      "sensor_id": "80116311- 1e27- 4fe1- 06c3- 9d6a75fe6f84",
      "metric_type": "in",
      "count": 0
    },
    ...
  ],
  "portal_metrics": [
    {
      "times": "2022- 01- 31T11: 00: 00",
      "portal_id": "287641f2- 47b5- 40c4- ac28- de2f53a3cf3c",
      "metric_type": "in",
      "count": 0,
      "modify_data": 0
    }
  ]
}
```

```
}  
]  
}
```

GET: /agent/service_mode/get_sum

запрос хэш-сумм метрик за промежуток времени. [wiki](#)

- **from** - начало промежутка. utc unixtime
- **to** - окончание промежутка. utc unixtime

Описание процессов

Автоматическая синхронизация метрик

- период сверки - 14 дней
- максимальный размер чанка - 7 дней (TODO)
- минимальный размер чанка - 1 день (TODO)

1. Начало итерации (по расписанию)
2. Сверка сумм
 1. запрос сумм в сервисе
 2. запрос сумм в коробке
 3. сравнение сумм в чанке
 1. при несовпадении сумм повторяем пункты 2.1, 2.2, 2.3 (бинарный поиск)
 2. при достижении минимального размера чанка - заносим промежуток в список **data_discrepancy**
3. Передача данных
 1. запрос данных в сервисе. параметры - промежутки из **data_discrepancy**
 2. передача данных в коробку
4. Окончание итерации

Синхронизация метрик по команде от сервиса

Идентично автоматической синхронизации метрик.

Исключение - период сверки, который берется из параметров команды

Передача измененных данных объектов из сервиса в коробочное издание

Передача измененных данных объектов из сервиса в коробку

Передача происходит через команду **`service_mode.sync_object_struct`**

Сервис

API

Команды:

Синхронизация структуры агента

POST: /control/client/{client_id}/device/command service_mode.sync_object_struct

```
{
  // Идентификатор не обязателен. Если не передается, сервис генерирует свой и возвращает в
  // ответе.
  "command_id": "1b38f303-a0cb-47ca-850c-450e4d9e4380",
  "device_id" : [], // пустой список
  "command": "service_mode.sync_object_struct",
  "params": {
    "object_struct": "Строка Base64"
  }
}
```

Синхронизация метрик

POST: /control/client/{client_id}/device/command service_mode.sync_metrics

// тело запроса:

```
"command_id": "1b38f303-a0cb-47ca-850c-450e4d9e4380", // идентификатор
команды
"device_id" : [], // пустой список
"command": "service_mode.sync_metrics", // тип команды
"params": {
  "from":1679300100, // начало
интервала
  "to": 1679300300 // конец
интервала
}
```

```
}  
}
```

ответ:

```
{  
  "error_flag": 0,  
  "messages": [],  
  "payload": {}  
}
```

Алгоритм выполнения:

- информация о команде записывается в БД в таблицу commands
- срабатывает триггер: {host}/agent/commands_check, отправляя текущую команду в соответствующий агент
- стандартный ответ, если запись в БД и триггер отработали успешно или ошибка если один из этапов завершился с ошибкой

Метрики за указанные период

GET: /agent/service_mode/get_metrics?from=&to=

// параметры:

"from" - начало запрашиваемого интервала в UNIX формате;
"to" - конец интервала в UNIX формате;

ответ:

```
"client_id": "de61e105-f743-44aa-b1ab-f799cd513229",  
"db_name": "DB_2",  
"from": 1679443000,  
"to": 1679529000,  
"metrics": [  
  {  
    "times": "2022-01-31T11:00:00",  
    "device_id": "a26c126b-7372-4123-a825-55d1526d1b1d",  
    "sensor_id": "80116311-1e27-4fe1-06c3-9d6a75fe6f84",  
    "metric_type": "in",  
    "count": 0
```



```

    }
  ],
  "portal_metrics": [
    {
      "times": "2022-01-31T11:00:00",
      "portal_id": "287641f2-47b5-40c4-ac28-de2f53a3cf3c",
      "metric_type": "in",
      "count": 0,
      "modify_data": 0
    }
  ]
}

```

алгоритм выполнения:

- сервис обращается к worker по gRPC
- worker отправляет запрос в СН, полученный результат отправляет обратно
- сервис формирует ответ и отправляет отправителю

Контрольные суммы за указанный период

GET: /agent/service_mode/get_sum?from=&to=

// параметры:

"from" - начало запрашиваемого интервала в UNIX формате;
 "to" - конец интервала в UNIX формате;

ответ:

```

{
  {
    "error_flag": 0,
    "messages": [],
    "payload": {
      "agent_id": "de61e105-f743-44aa-b1ab-f799cd513229",
      "from": 1679300100,
      "to": 1679300300,
      "hash_device":
"831e7c6c89053b74840fbc72e972d10c24757d4225784e1c7ab25a5e08199e04",
      "hash_portal":
"ffa45493d81296512e0eea5a80c456e5938159b0f5aba5d59318c60d8df6a489"
    }
  }
}

```

```
}  
}
```

алгоритм выполнения:

- сервис обращается к worker по grpc
- worker отправляет запрос в СН, полученный результат отправляет обратно
- сервис рассчитывает хеш-сумму, формирует ответ и отправляет результат отправителю

Сравнение хэш-сумм метрик за период (СТАРЫЙ НЕ ИСПОЛЬЗОВАТЬ)

Для сравнения хэш-сумм используется функция хэширования SHA256. Хэшируется JSON строка выборки данных за указанный период. Структура данных для метрик по устройствам и метрикам по переходам указана ниже.

Обязательно условия выборки данных:

1. Сортировка:

1. Для переходов

1. times - по возрастанию
2. portal_id - по возрастанию
3. metric_type - по возрастанию

2. Для устройств

1. times - по возрастанию
2. device_id - по возрастанию
3. sensor_id - по возрастанию
4. metric_type - по возрастанию

2. Порядок полей:

1. Для переходов

1. times
2. portal_id
3. metric_type
4. count
5. modify_data

2. Для устройств

1. times
2. device_id
3. sensor_id
4. metric_type
5. count

3. В строке JSON должны быть полностью исключены все символы кроме значимых (исключить пробелы, переносы строк и т.п.).
4. Формат даты ISO, пример "2019-03-15T00:00:00".

Тело ответа на запрос хэш-суммы за период:

```
{
  "error_flag": 0,
  "messages": [],
  "payload": {
    "agent_id": "de61e105- f743- 44aa- b1ab- f799cd513229",
    "from": 1679300100,
    "to": 1679300300,
    "hash_device":
"831e7c6c89053b74840fbc72e972d10c24757d4225784e1c7ab25a5e08199e04",
    "hash_portal": " ffa45493d81296512e0eea5a80c456e5938159b0f5aba5d59318c60d8df6a489"
  }
}
```

Пример структуры для переходов:

```
[
  {
    "times": "2022- 01- 31T11: 00: 00",
    "portal_id": "75c1f93c- 6d8f- 4bf0- bb1c- f5852e178b88",
    "metric_type": "in",
    "count": 15,
    "modify_data": 0
  },
  {
    "times": "2022- 01- 31T11: 00: 00",
    "portal_id": "75c1f93c- 6d8f- 4bf0- bb1c- f5852e178b88",
    "metric_type": "out",
  }
]
```

```

    "count": 11,
    "modify_data": 0
  },
  {
    "times": "2022-01-31T11:00:00",
    "portal_id": "75c1f93c-6d8f-4bf0-bb1c-f5852e178b88",
    "metric_type": "pass",
    "count": 11,
    "modify_data": 0
  },
  {
    "times": "2022-01-31T11:00:00",
    "portal_id": "75c1f93c-6d8f-4bf0-bb1c-f5852e178b88",
    "metric_type": "unknown",
    "count": 11,
    "modify_data": 0
  }
]

```

Пример структуры для устройств:

```

[
  {
    "times": "2022-01-31T11:00:00",
    "device_id": "75c1f93c-238f-4bf0-bb1c-f5852e178b88",
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",
    "metric_type": "in",
    "count": 15
  },
  {
    "times": "2022-01-31T11:00:00",
    "device_id": "75c1f93c-238f-4bf0-bb1c-f5852e178b88",
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",
    "metric_type": "out",
    "count": 11
  },
  {
    "times": "2022-01-31T11:00:00",
    "device_id": "75c1f93c-238f-4bf0-bb1c-f5852e178b88",
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",

```

```
    "metric_type": "pass",  
    "count": 11  
  },  
  {  
    "times": "2022-01-31T11:00:00",  
    "device_id": "75c1f93c-238f-4bf0-bb1c-f5852e178b88",  
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",  
    "metric_type": "unknown",  
    "count": 11  
  }  
]
```

API 1С

GET: /agent/service_mode/get_sum

запрос хэш-сумм метрик за промежуток времени (sha256). Подробнее про алгоритм хэширование в [этой статье](#).

Параметры:

- from - начало промежутка. utc unixtime
- to - окончание промежутка. utc unixtime

Ответ:

```
{
  "error_flag": 0,
  "messages": [],
  "payload": {
    "agent_id": "de61e105- f743- 44aa- b1ab- f799cd513229",
    "from": 1679300100,
    "to": 1679300300,
    "hash_device":
"831e7c6c89053b74840fbc72e972d10c24757d4225784e1c7ab25a5e08199e04",
    "hash_portal": "ffa45493d81296512e0eea5a80c456e5938159b0f5aba5d59318c60d8df6a489"
  }
}
```

POST: /agent/service_mode/object_struct

Отправка измененных данных объектов из сервиса в коробочное издание.

```
{
  "object_struct": "BASE64 строка"
}
```

POST: /agent/service_mode/metrics

Отправка метрик из сервиса в коробку. Предполагается пересылка без изменения формата

Пример тела запроса

```
{
  "client_id": "de61e105-f743-44aa-b1ab-f799cd513229",
  "db_name": "DB_2",
  "from": 1679443000,
  "to": 1679529000,
  "metrics": [
    ...,
    {
      "times": "2022-01-31T11:00:00",
      "device_id": "a26c126b-7372-4123-a825-55d1526d1b1d",
      "sensor_id": "80116311-1e27-4fe1-06c3-9d6a75fe6f84",
      "metric_type": "in",
      "count": 0
    },
    ...
  ],
  "portal_metrics": [
    {
      "times": "2022-01-31T11:00:00",
      "portal_id": "287641f2-47b5-40c4-ac28-de2f53a3cf3c",
      "metric_type": "in",
      "count": 0,
      "modify_data": 0
    }
  ]
}
```


Сравнение хэш-сумм метрик за период

Для сравнения хэш-сумм используется функция хэширования SHA256. Хэшируется JSON строка выборки данных за указанный период. Структура данных для метрик по устройствам и метрикам по переходам указана ниже.

Обязательно условия выборки данных:

1. Сортировка:

1. Для переходов

1. portal_id - по возрастанию
2. metric_type - по возрастанию

2. Для устройств

1. device_id - по возрастанию
2. sensor_id - по возрастанию
3. metric_type - по возрастанию

2. Порядок полей:

1. Для переходов

1. portal_id
2. metric_type
3. count
4. modify_data

2. Для устройств

1. device_id
2. sensor_id
3. metric_type
4. count

3. В строке JSON должны быть полностью исключены все символы кроме значимых (исключить пробелы, переносы строк и т.п.).

Тело ответа на запрос хэш-суммы за период:

```
{
  "error_flag": 0,
  "messages": [],
  "payload": {
    "agent_id": "de61e105-f743-44aa-b1ab-f799cd513229",
    "from": 1679300100,
    "to": 1679300300,
    "hash_device":
"831e7c6c89053b74840fbc72e972d10c24757d4225784e1c7ab25a5e08199e04",
    "hash_portal": "ffa45493d81296512e0eea5a80c456e5938159b0f5aba5d59318c60d8df6a489"
  }
}
```

Пример структуры для переходов:

```
[
  {
    "portal_id": "75c1f93c-6d8f-4bf0-bb1c-f5852e178b88",
    "metric_type": "in",
    "count": 15,
    "modify_data": 0
  },
  {
    "portal_id": "75c1f93c-6d8f-4bf0-bb1c-f5852e178b88",
    "metric_type": "out",
    "count": 11,
    "modify_data": 0
  },
  {
    "portal_id": "75c1f93c-6d8f-4bf0-bb1c-f5852e178b88",
    "metric_type": "pass",
    "count": 11,
    "modify_data": 0
  },
  {
    "portal_id": "75c1f93c-6d8f-4bf0-bb1c-f5852e178b88",
    "metric_type": "unknown",
    "count": 11,

```

```
    "modify_data": 0
  }
]
```

Пример структуры для устройств:

```
[
  {
    "device_id": "75c1f93c-238f-4bf0-bb1c-f5852e178b88",
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",
    "metric_type": "in",
    "count": 15
  },
  {
    "device_id": "75c1f93c-238f-4bf0-bb1c-f5852e178b88",
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",
    "metric_type": "out",
    "count": 11
  },
  {
    "device_id": "75c1f93c-238f-4bf0-bb1c-f5852e178b88",
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",
    "metric_type": "pass",
    "count": 11
  },
  {
    "device_id": "05b0b047-013f-4b88-8534-b2ddecf80384",
    "sensor_id": "45c1f93c-6d8f-4650-bb1c-f5852e156b78",
    "metric_type": "unknown",
    "count": 11
  }
]
```