

# 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 gpc
- worker отправляет запрос в СН, полученный результат отправляет обратно
- сервис рассчитывает хеш-сумму, формирует ответ и отправляет результат отправителю

---

Revision #40

Created 6 July 2023 08:40:54 by Николай

Updated 7 November 2024 14:13:32 by Николай