

ООО «НТК Интерфейс»

# IP-шлюз ARIS-SCADA

Инструкция по использованию

Екатеринбург 2013

# СОДЕРЖАНИЕ

Введение.....	3
1 Принцип работы.....	3
2 Программа «IP-шлюз ARIS-SCADA».....	3
2.1 Аргументы командной строки запуска .....	4
2.2 Примеры аргументов при запуске приложения .....	5
2.3 Примеры аргументов при запуске внешней задачи .....	6
2.4 «Белый список» адресов.....	6
3 Формат символьных запросов и ответов .....	6
3.1 Проверка работоспособности шлюза .....	7
3.2 Текущие данные .....	7
3.3 Ретроспектива измерений .....	10
3.4 События .....	11
3.4.1 Архив событий .....	12
3.4.2 Последние события .....	14
3.5 База данных телеметрии.....	15
3.5.1 Выборка каналов.....	15
3.5.2 Выборка КП.....	15
3.5.3 Выборка объектов.....	16
3.6 Команды воздействия на данные .....	17
3.6.1 Установка состояния сигнала .....	18
3.6.2 Переключение состояния сигнала.....	18
3.6.3 Квитирование сигнала.....	19
3.6.4 Установка значения измерения .....	19
Приложение А Расшифровки сообщений об ошибках шлюза .....	20

# Введение

Настоящий документ является руководством по организации удаленного доступа к данным Сервера ARIS-SCADA с помощью IP-протокола.

## 1 Принцип работы

Доступ к данным Сервера ARIS-SCADA организуется следующим образом:

- На компьютере с операционной системой семейства Windows NT устанавливается и настраивается программа «IP-шлюз ARIS-SCADA» (далее шлюз). Установить шлюз можно или непосредственно на компьютере с Сервером ARIS-SCADA, или на удаленном, находящемся в одной подсети с сервером.
- Удаленный пользователь-клиент с помощью TCP-сокета организует сеанс связи со шлюзом, отправляет символьный запрос и получает ответ, содержащий значения запрошенной информации.
- Частным случаем работы клиента шлюза являются установленные на веб-сервере скрипты, осуществляющие коммуникацию со шлюзом и отображающие данные телеметрии и события. Это позволяет пользователю удаленно наблюдать информацию по сети Интернет, используя только программу-браузер и любые доступные каналы связи, включая сети GPRS.

## 2 Программа «IP-шлюз ARIS-SCADA»

Программа состоит из следующих компонентов:

- OikIPGate.exe – исполняемый файл самостоятельного запуска;
- OikIPGateET.exe – файл для запуска в качестве внешней задачи сервера;
- white\_list.txt – белый список IP-адресов пользователей;
- Logs/ – директория для хранения лог-файлов программы;
- cfsshare.dll – основная связная библиотека сервера ARIS-SCADA;
- tmconn.dll – библиотека связи с сервером телемеханики.

Программа может работать как внешняя задача сервера ARIS-SCADA (рекомендуемый вариант), так и автономно на любом удаленном компьютере. Для этих целей используется два разных исполняемых файла.

Автономная программа запускается из командной строки Windows или с помощью ярлыка. Программа имеет консольный интерфейс. Снимок экрана работающей программы приведен на рисунке 1.

После запуска программа (в любом режиме запуска) в фоновом режиме постоянно прослушивает указанный в настройках порт. При подключении клиента посредством TCP-сокета по данному порту шлюз ожидает появления строки запроса, организует сеанс связи с Сервером, принимает требуемые данные телеметрии, формирует ответ и отправляет его клиенту, закрывая соединение.



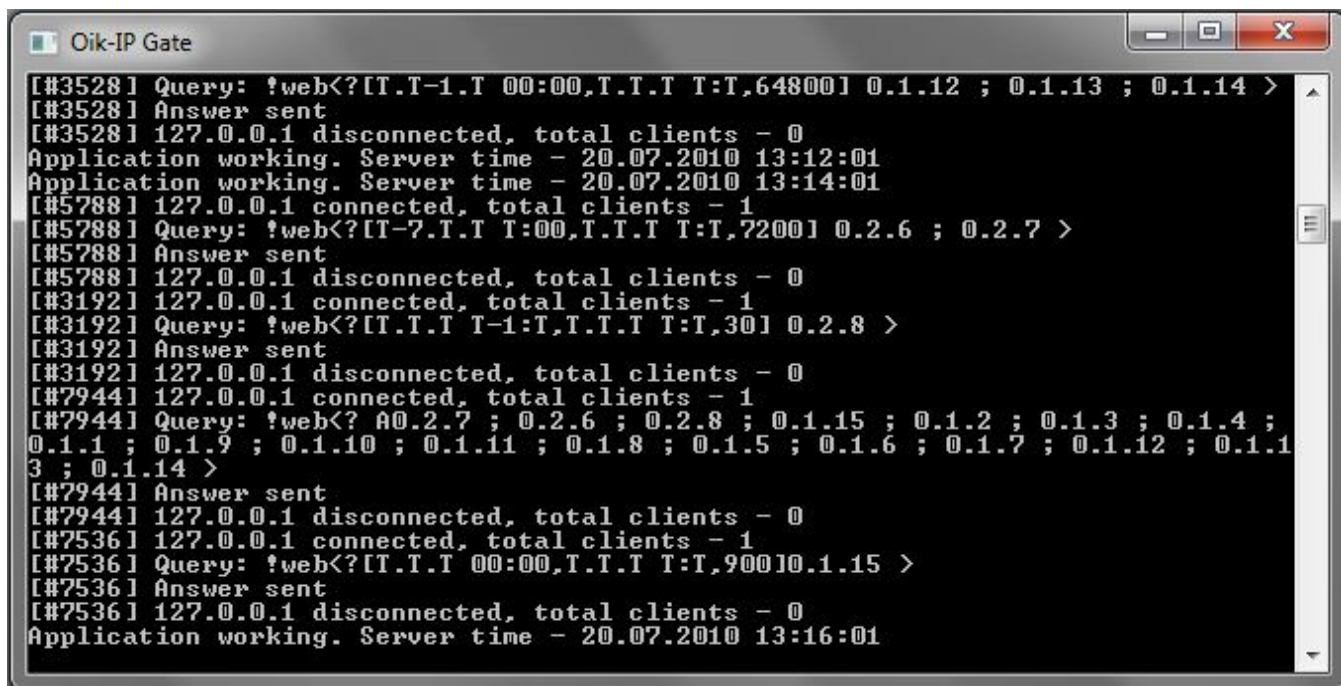
The screenshot shows a console window titled "Oik-IP Gate". It displays a series of log messages in a monospaced font. The messages include connection status updates (e.g., "127.0.0.1 disconnected, total clients - 0"), application status reports ("Application working. Server time - 20.07.2010 13:12:01"), and query-response pairs (e.g., "[#3528] Query: !web<?[T.T.T T:00,T.T.T T:T,64800] 0.1.12 ; 0.1.13 ; 0.1.14 >"). The logs show multiple clients connecting and disconnecting, with the total client count fluctuating between 0 and 1. The server time is consistently 20.07.2010.

Рисунок 1 – IP-шлюз в консольном интерфейсе

Обратите внимание, если на компьютере установлен брандмауэр, вероятно потребуется добавить программу в исключения и разрешить как прослушивание заданного порта, так и общение с клиентами по этому же порту (либо со всеми клиентами, либо ограничить по адресам с требуемыми).

При необходимости завершить работу программы рекомендуется воспользоваться нажатием сочетания клавиш клавиатуры Ctrl+C. При простом закрытии окна может появляться сообщение о невозможности завершения, в этом случае предупреждение следует игнорировать, через 5 секунд окно нормально закроется.

## 2.1 Аргументы командной строки запуска

— /server=...

Имя компьютера, на котором установлен Сервер ARIS-SCADA.

Если IP-шлюз расположен на той же машине, что и сервер, то указывается значение «.» (точка). В противном случае требуется указать алиас сервера, взятый из программы cfssetup.exe.

При запуске как внешняя задача параметр указывать не обязательно.

— /pipe=...

Имя сервера динамических данных (по умолчанию TMS).

— /port=...

TCP-порт для работы шлюза.

Чтобы шлюз смог начать прослушивание, порт на данной машине не должен быть занят другим приложением.



— /timeout=...

Время (в секундах) ожидания текста запроса от клиента после соединения. Если запрос не был получен в заданный промежуток, то соединение будет принудительно закрыто.

— /length=...

Максимальная длина запроса (в байтах) от клиента. В случае превышения ответ шлюза будет состоять из предупредительного сообщения.

— /threads=...

Максимальное число одновременных соединений к шлюзу. Если в какой-то момент число соединений достигнет максимального, то новые запросы на соединения будут мгновенно отклоняться до появления свободного места.

— /log=...

В случае указания значения «1» шлюз будет записывать подробные логи действий. Все логи сохраняются в директории «Logs». При запуске в качестве внешней задачи параметр игнорируется, и логи в текстовые файлы не записываются. Для просмотра сообщений программы можно пользоваться встроенной трассировкой Сервера ARIS-SCADA с установленным флагом «Отладка».

Допускаются значения 0 (не писать логи) и 1 (писать).

— /commandkey=...

Текстовый пароль для выполнения команд, воздействующих на данные сервера (см. раздел 3.6).

Если пароль не указан, то выполнение команд будет запрещено.

Все параметры являются опциональными. Ограничения и значения по умолчанию указаны далее в таблице 1.

Таблица 1 – Аргументы командной строки шлюза

Свойство	По умолчанию	Диапазон допустимых значений
/server=	.	
/pipe=	TMS	
/port=	979	1 – 65535
/timeout=	5	1 – 60
/length=	8192	1 – 8192
/threads=	5	1 – 32
/log=	0	0 – 1
/commandkey=		

## 2.2 Примеры аргументов при запуске приложения

*OikIPGate.exe*

Так как аргументы не были заданы, все свойства будут приняты по умолчанию.



```
OikIPGate.exe /server=. /pipe=TMS /port=81 /timeout=5 /length=1024 /threads=30 /log=0
```

Соединение с сервером на локальной машине, сервер TMS, программа занимает порт 81, таймаут запроса 5 секунд, длина запроса ограничена одним килобайтом, число одновременных соединений не превышает 30, запись логов не требуется.

## 2.3 Примеры аргументов при запуске внешней задачи

```
/port=81 /timeout=5 /length=1024 /threads=30
```

Программа будет запущена на заданном сервере, займет порт 81, таймаут запроса 5 секунд, длина запроса ограничена одним килобайтом, число одновременных соединений не превышает 30.

```
/port=20113
```

Программа будет запущена на заданном сервере и займет порт 20113. Остальные параметры – по умолчанию.

## 2.4 «Белый список» адресов

Существует возможность задания «белого списка» адресов: программа будет принимать запросы только, если IP-адрес подключающегося клиента совпадает с одним из адресов списка. Список хранится в файле *white\_list.txt* в корневой папке программы, каждый адрес расположен в отдельной строке.

Максимально допустимое число адресов – 16. В случае, если файл отсутствует или является пустым, шлюз будет обрабатывать все запросы.

Пример содержимого файла белого списка:

```
10.0.0.10  
10.0.0.11  
10.0.0.256  
81.91.36.3  
81.91.36.5
```

В данном случае будут обрабатываться запросы только с указанных пяти адресов, связь с другими машинами будет сразу прерываться.

## 3 Формат символьных запросов и ответов

Как запросы клиентов шлюзу, так и ответы шлюза подчиняются формату JSON. В общем виде запрос выглядит следующим образом:

```
{"id": "<идентификатор>", "query": "<тип запроса>"}
```

где идентификатор – опциональный идентификатор клиента, выполняющего запрос. При ответе шлюз вернет этот же идентификатор.



В зависимости от типа текст запроса может содержать дополнительные свойства. Описание типов запросов приведено далее в документе.

Подробнее о формате JSON можно прочитать на специализированных сайтах в сети Интернет, например, <http://www.json.org/json-ru.html>. Ввиду эффективности и распространенности формата большинство современных языков имеют встроенные или сторонние средства для сериализации и десериализации данных, что облегчает возможную разработку приложений для работы с данными шлюза.

### 3.1 Проверка работоспособности шлюза

Запрос для проверки работоспособности шлюза:

```
{"query": "test"}
```

Здесь и далее в документе для удобства понимания никогда не указывается идентификатор клиента. В случае указания запрос мог выглядеть так:

```
{"id": "test-client", "query": "test"}
```

Ответ шлюза, если всё в порядке, будет состоять из ключа успешности "ok", а также даты и времени сервера в ключе "serverTime". Ответ может выглядеть так:

```
{"ok": "1", "serverTime": "04.02.2013 13:33:30"}
```

В случае возникновения ошибки ответ будет состоять из ключа ошибки и краткого сообщения о типе возникшей проблемы:

```
{"error": "..."} 
```

Расшифровки ответов с ошибками шлюза приведены в приложении А.

В случае полного отсутствия ответа вероятнее всего соединение не было установлено. Следует проверить настройки (как на шлюзе, так и на клиенте):

- Должны быть указаны корректные адреса и совпадающие порты;
- Если для шлюза указан «белый список», то адрес клиента должен в нем присутствовать. Этот момент можно отследить в логах шлюза (непосредственно в консоли приложения или в отладочной трассировке, если шлюз запущен как внешняя задача) – если подключающийся клиент не в «белом списке», выводится соответствующее сообщение;
- Сеанс общения может быть заблокирован брандмауэром с любой из сторон. Следует отследить наличие всплывающих сообщений брандмауэра или проверить системные логи. Для исправления данной проблемы следует добавить программы и/или заданный адрес и порт в исключения.

### 3.2 Текущие данные

Запрос имеет следующую структуру:

```
{  
  "query": "instant",  
}
```



```

"ts": ["ID1", "ID2", "...", "IDn"],
"ti": ["ID1", "ID2", "...", "IDn"],
"tii": ["ID1", "ID2", "...", "IDn"]
}

```

где:

"ts" – ключ для указания списка запрашиваемых сигналов;

"ti" – ключ для указания списка запрашиваемых измерений;

"tii" – ключ для указания списка запрашиваемых интегральных измерений;

ID<sub>n</sub> – адрес параметра, в формате Канал.КП.Объект, например, 13.127.1.

Каждый адрес в запросе должен быть заключен в кавычки.

В последнем примере, а также некоторых последующих, тексты запросов и ответов для удобства просмотра форматируются символами переноса строки. При реальном использовании шлюза так оформлять запрос не нужно.

С помощью одной команды допускается запрашивать как только однотипные данные (например, сигналы), так и все типы сразу. При этом адресация данных независима для каждого типа.

Например, для запроса ряда измерений с блоком адресов 0.1.1 – 0.1.6 используется следующий запрос:

```

{"query": "instant", "ti": ["0.1.1", "0.1.2", "0.1.3", "0.1.4", "0.1.5", "0.1.6"]}

```

Для запроса двух сигналов, четырех измерений и одного интегрального измерения может быть использована следующая команда:

```

{
  "query": "instant",
  "ts": ["0.100.1", "0.100.5"],
  "ti": ["0.2.6", "0.2.7", "0.2.8", "0.2.9"],
  "tii": ["0.1.1"]
}

```

Обратите внимание, одной разрешается запросить не более 1000 параметров каждого типа. В противном случае ответ шлюза будет состоять из сообщения об ошибке (см. приложение А).

Тело ответа шлюза формируется из списка значений запрошенных параметров с флагами. Для каждого типа данных формируется отдельный список. Порядок следования переменных внутри списка совпадает с порядком в запросе:

```

[
  {"v": "Знач1", "f": "ФЛ1"},
  {"v": "Знач2", "f": "ФЛ2"},
  ...,
  {"v": "ЗначN", "f": "ФЛN"}
]

```

Для интегральных измерений дополнительно указывается нагрузка с помощью ключа "l", как показано ниже:





```
[
  {"v": "Знач1", "l": "Нагр1", "f": "ФЛ1"},
  ...
  {"v": "ЗначN", "l": "НагрN", "f": "ФЛN"}
]
```

Обратите внимание, в ответе шлюза все значения заключены в кавычки (являются строками, а не числами), при этом разделителем целой и дробной частей является символ запятой.

Флаги задаются строкой из набора символов. Типы символов и их расшифровки представлены в таблице 2.

Таблица 2 – флаги параметров

Флаг	Расшифровка
E	Недостоверность аппаратная
R	Заблокировано оператором
M	Установлено вручную
Z	Замена/резерв
N	Отличие от нормы (для сигналов)
U	Неквитированность (для сигналов)
1	Признак 1
2	Признак 2
3	Признак 3
4	Признак 4

В случае отсутствия флагов у параметра, ключ "f" для данного параметра в ответе не упоминается.

Если указанный параметр не описан на сервере, то для него шлюз возвращает значение «No» без флагов.

Примеры запросов текущих данных и возможных ответов шлюза:

— Запрос шести измерений, последнее из которых не описано на сервере:

```
{"query": "instant", "ti": ["0.1.1", "0.1.2", "0.1.3", "0.1.4", "0.1.5", "0.1.6"]}
```

```
{"ti": [{"v": "123,34"}], [{"v": "10002"}], [{"v": "1234,56789", "f": "E"}], [{"v": "1385,1"}], [{"v": "31415,926"}], [{"v": "No"}]}
```

— Запрос двух сигналов, четырех измерений и одного интегрального измерения:

```
{
  "query": "instant",
  "ts": ["0.100.1", "0.100.5"],
  "ti": ["0.2.6", "0.2.7", "0.2.8", "0.2.9"],
  "tii": ["0.1.1"]
}
```



```

{
  "ts": [{ "v": "0" }, { "v": "1", "f": "N" } ],
  "ti": [{ "v": "100500", "f": "R" }, { "v": "271828,4590" }, { "v": "12,34" }, { "v": "56,78" } ],
  "tii": [{ "v": "144187,00", "l": "600,00" } ]
}

```

### 3.3 Ретроспектива измерений

При запросе ретроспективы указываются следующие параметры: время начала выборки, время конца, шаг выборки и список измерений. Число одновременно запрашиваемых измерений не должно превышать 5.

Запрос имеет следующую структуру:

```

{
  "query": "retro",
  "start": "<ВремяНачала>",
  "end": "<ВремяКонца>",
  "step": <ШагВыборки>,
  "ti": ["ID1", "ID2", "...", "ID5"]
}

```

Формат метки времени для начала и конца выборки:

ДД.ММ.ГГГГ ЧЧ:ММ

Присутствует возможность задавать относительное значение единицы времени, с помощью символа Т, который равнозначен значению текущего момента. Пример корректных меток времени приведен в таблице 3.

Таблица 3 – примеры корректных меток времени ретроспективы

Метка	Описание
22.04.2012 9:08	Прямое указание времени
T.T.T T:T	Текущий момент времени
T.T.T 0:0	Начало текущих суток
T-1.T.T 0:0	Начало предыдущих суток
T.T.T T:0	Начало текущего часа
1.T.T 0:0	Начало текущего месяца
T.T.T T-1:T	Минус один час относительно текущего момента
T-7.T.T T:T	Минус неделя относительно текущего момента

Шаг выборки измеряется в секундах и задается одним числом. Например, для задания шага в 30 минут следует указать значение 1800 (30 × 60).

Максимальное допустимое значение шага – 2678400 (31 сутки).

Задание списка измерений не отличается от списка при запросе текущих данных (см. раздел 3.2).

Обратите внимание, в ретроспективе выборки одного измерения должно быть не более 1000 значений, а число измерений в одном запросе не должно превышать 5. В



противном случае ответ шлюза будет состоять из сообщения об ошибке (см. приложение А).

Ответ шлюза аналогичен ответу при запросе текущих данных: в ключе "ti" идет перечисление списка значений с флагами по порядку от начального времени до конечного с заданным шагом, значение времени элементов выборки при этом не передается.

Примеры запросов ретроспективы измерений и возможных ответов шлюза:

- Запрос ретроспективы измерения с адресом 0.1.1 с начала суток до текущего момента с шагом выборки 15 минут:

```
{
  "query": "retro",
  "start": "T.T.T 0:0",
  "end": "T.T.T T:T",
  "step": 900,
  "ti": ["0.1.1"]
}
```

```
{ "ti": [
  [{"v": "123,34"}], [{"v": "1234,56789", "f": "E"}], ..., [{"v": "31415,926"}]
}
```

- Запрос ретроспективы измерений в блоке адресов 0.5.1 – 0.5.3, архив в конкретный промежуток времени с шагом выборки в 5 минут:

```
{
  "query": "retro",
  "start": "13.07.2010 00:00",
  "end": "13.07.2010 10:10",
  "step": 300,
  "ti": ["0.5.1", "0.5.2", "0.5.3"]
}
```

```
{ "ti": [
  [{"v": "223,340"}, {"v": "223,340", "f": "E"}, {"v": "221,110"}],
  [{"v": "212,7"}, {"v": "212,7", "f": "E"}, {"v": "212,8"}],
  [{"v": "220,1"}, {"v": "220,1", "f": "E"}, {"v": "219,8"}]
}
```

### 3.4 События

Шлюз обеспечивает поддержку двух видов запроса событий:

- архив событий с указанием времени начала и конца выборки;
- последние события с указанием количества событий.



При любом из запросов ответ будет содержать список событий внутри ключа "events". Каждое из событий характеризуется следующими данными:

- уровень важности события:
  - 0 – оперативного состояния;
  - 1 – предупредительные 1;
  - 2 – предупредительные 2;
  - 3 – аварийные;
- время;
- адрес связанного параметра;
- имя связанного параметра;
- тип;
- новое состояние.

Описание каждого события подчиняется следующему формату:

```
{  
  "imp": <Важность>,  
  "dat": "<Время>"  
  "adr": "<АдресПараметра>",  
  "n": "<ИмяПараметра>",  
  "t": "<Тип>",  
  "st": "<Состояние>"  
}
```

Например:

```
{  
  "imp": 1,  
  "dat": "19.04.2012 09:47:48.849",  
  "adr": "#TC22:13:1",  
  "n": "Демо АТ-1 6 кВ В-6",  
  "t": "Выключатель",  
  "st": "Включен РУЧН."  
}
```

Обратите внимание, запрос событий доступен для сервера ARIS-SCADA с датой компоновки не ранее февраля 2012 года. Если шлюз подключен к серверу более старой версии, то ответ будет состоять из сообщения об отсутствии поддержки событий (см. приложение А).

### 3.4.1 Архив событий

При запросе архива событий указываются следующие параметры: время начала выборки, время конца, а также опционально фильтры по важности и типу событий и адрес сигнала (например, для запроса ретроспективы переключений аппарата).

Текст запроса имеет следующую структуру:

```
{  
  "query": "events-archive",  
  "start": "<ВремяНачала>",
```



```

        "end": "<ВремяКонца>",
        "imp": <МаскаВажностиСобытий>,
        "type": <МаскаТипаСобытий>
        "addr": "<АдресСигнала>"
    }

```

Метки времени начала и конца выборки полностью аналогичны меткам запроса архива измерений (см. раздел 3.3).

Маски важности и типа – битовые маски для задания фильтра событий. Описание флагов масок приведено в таблицах 4 и 5.

Таблица 4 – флаги маски важности событий

Флаг	Описание
1	Уровень 0 (оперативного состояния)
2	Уровень 1 (предупредительные 1)
4	Уровень 2 (предупредительные 2)
8	Уровень 3 (аварийные)
0	Все уровни важности

Таблица 5 – флаги маски типов событий

Флаг	Описание
1	Изменение сигналов
2	Выходы измерений за уставки
4	Команды управления
8	Квотирование оператором
16	Ручная установка сигналов
32	Ручная установка измерений
0	Все типы

Для задания фильтра следует выбрать требуемые флаги и сложить их, результат сложения и будет маской.

Например, маска типов ручной установки сигналов и измерений – 48 (16 + 32).

Маска важности всех событий кроме оперативного состояния – 14 (2 + 4 + 8).

Если фильтр не требуется, в качестве маски можно задать значение 0 или вообще не указывать ключ фильтра в запросе.

Пример запроса архива изменения сигналов (маска 1) с уровнем важности «предупредительные 2» и «аварийные» (маска 4 + 8) за фиксированный промежуток времени:

```

{
    "query": "events-archive",
    "start": "02.04.2012 17:00",
    "end": "02.04.2012 18:00",
    "imp": 12,
    "type": 1
}

```

**Возможный ответ шлюза:**

```

      {"events": [
{"imp":3,"dat":"02.04.2012 17:06:21.888","adr":"#TC20:1:1","n":"Демо ВЛ 220 кВ Бор В-220","t":"Выключатель","st":"Отключен РУЧН."},
{"imp":3,"dat":"02.04.2012 17:16:23.211","adr":"#TC20:1:1","n":"Демо ВЛ 220 кВ Бор В-220","t":"Выключатель","st":"Включен РУЧН."}
]}

```

**Пример запроса архива всех событий за последние сутки:**

```

{
  "query": "events-archive",
  "start": "T-1.T.T T:T",
  "end": "T.T.T T:T "
}

```

**Возможный ответ шлюза:**

```

      {"events": [
{"imp":3,"dat":"19.04.2012 10:22:51.316","adr":"#TC20:1:1","n":"Демо ВЛ 220 кВ Бор В-220","t":"Выключатель","st":"Отключен"},
{"imp":3,"dat":"19.04.2012 10:23:08.635","adr":"#TC20:1:1","n":"Демо ВЛ 220 кВ Бор В-220","t":"Выключатель","st":"Включен"},
{"imp":2,"dat":"19.04.2012 10:41:14.835","adr":"#TC20:1:5","n":"Демо ВЛ 220 кВ Бор ЗН ШР-220","t":"Разъединитель","st":"Включен"},
{"imp":3,"dat":"19.04.2012 10:41:29.171","adr":"#TC20:1:1","n":"Демо ВЛ 220 кВ Бор В-220","t":"Выключатель","st":"Отключен"},
{"imp":2,"dat":"19.04.2012 10:41:31.131","adr":"#TC20:1:5","n":"Демо ВЛ 220 кВ Бор ЗН ШР-220","t":"Разъединитель","st":"Отключен"},
{"imp":0,"dat":"19.04.2012 10:47:55.020","adr":"#TC0:0:0","n":"Общее квитирование ТС","t":"Квитирование","st":""}
]}

```

### 3.4.2 Последние события

Запрос последних по времени событий выглядит следующим образом:

```

{"query": "events-latest", "quantity": <количество>}

```

В списке ответа шлюза будет заданное количество последних событий сервера. Обратите внимание, что количество запрашиваемых событий не может превышать 2000.

**Пример запроса 5 последних событий:**

```

{"query": "events-latest", "quantity": 5}

```

**Возможный ответ шлюза:**

```

      {"events": [
{"imp":1,"dat":"19.04.2012 09:47:48.849","adr":"#TC22:13:1","n":"Демо АТ-1 6 кВ В-6","t":"Выключатель","st":"Включен РУЧН."},
{"imp":3,"dat":"19.04.2012 10:22:51.316","adr":"#TC20:1:1","n":"Демо ВЛ 220 кВ Бор В-220","t":"Выключатель","st":"Отключен"},
{"imp":1,"dat":"19.04.2012 10:23:02.467","adr":"#TC20:1:3","n":"Демо ВЛ 220 кВ Бор ЛР-220","t":"Разъединитель","st":"Отключен"},
{"imp":3,"dat":"19.04.2012 10:23:08.635","adr":"#TC20:1:1","n":"Демо ВЛ 220 кВ Бор В-220","t":"Выключатель","st":"Включен"}
]}

```



```
220", "t": "Выключатель", "st": "Включен"},
{"imp": 0, "dat": "19.04.2012 10:23:12.450", "adr": "#TC20:1:4", "n": "Демо ВЛ 220 кВ Бор ОР-
220", "t": "Разъединитель", "st": "Включен"}
]}
```

### 3.5 База данных телеметрии

Запросы к базе данных позволяют осуществлять навигацию по дереву телеметрии. Навигация трехуровневая: выборка каналов, выборка КП внутри заданного канала, выборка объектов внутри КП.

Каждый уровень имеет свою структуру запроса. Далее в документе приведено подробное описание и примеры для всех уровней.

Формат ответа шлюза не зависит от того, какой уровень элементов был запрошен. Кроме списка элементов, включающего в себя индекс и наименование каждого из элементов, в ответе выводится общее количество элементов, а также индекс последнего элемента.

```
{
  "total": <ОбщееКоличество>,
  "lastId": <ИндексПоследнегоЭлемента>,
  "items": [
    {"n": "ИмяЭлемента1", "id": ИндексЭлемента1},
    {"n": "ИмяЭлемента2", "id": ИндексЭлемента2},
    ...,
    {"n": "ИмяЭлементаN", "id": ИндексЭлементаN}
  ]
}
```

#### 3.5.1 Выборка каналов

Для запроса списка каналов следует воспользоваться следующей командой:

```
{"query": "tms-ch"}
```

Пример ответа шлюза:

```
{
  "total": 6,
  "lastId": 24,
  "items": [
    {"n": "Внутренние данные", "id": 0},
    {"n": "Демо ОРУ 220 кВ", "id": 20},
    {"n": "Демо ОРУ 110 кВ", "id": 21},
    {"n": "Демо КРУН 6 кВ", "id": 22},
    {"n": "Демо ЩСН 0,4 кВ", "id": 23},
    {"n": "Демо служ. инф.", "id": 24}
  ]
}
```

#### 3.5.2 Выборка КП

В запросе списка КП добавляется индекс родительского канала:

```
{"query": "tms-rtu", "ch": <канал>}
```



Например, запрос списка КП канала 20:

```
{"query": "tms-rtu", "ch": 20}
```

Возможный ответ шлюза:

```
{
  "total":8,
  "lastId":8,
  "items":[
    {"n":"ВЛ 220 кВ Бор - Демо","id":1},
    {"n":"ОВ 220 кВ","id":2},
    {"n":"АТ-1 220 кВ","id":3},
    {"n":"ВЛ 220 кВ Анна - Демо","id":4},
    {"n":"АТ-2 220 кВ","id":5},
    {"n":"ТН-1 220 кВ","id":6},
    {"n":"ТН-2 220 кВ","id":7},
    {"n":"ТН ОСШ 220 кВ","id":8}
  ]
}
```

### 3.5.3 Выборка объектов

Для запроса списка объектов (сигналов, измерений или интегральных измерений) требуется указывать индексы канала и КП.

Опционально можно также указать начальный индекс объекта, в этом случае объекта с индексами ниже не появятся в выборке. Необходимость такого параметра вызвана ограничением размера выборки – в одной команде можно запросить не более 255 объектов.

Тогда для того, чтобы гарантированно получить весь список, следует запросить список без начального индекса, затем с помощью значений общего количества объектов и последнего индекса запроса сформировать и выполнить дополнительные запросы.

Текст запроса выглядит следующим образом:

```
{"query": "tms-<Тип>", "ch": <Канал>, "rtu": <Кп>, "from": <НачальныйИндекс>}
```

где тип:

ts – для сигналов;

ti – для измерений;

tii – для интегральных измерений.

Например, чтобы запросить сигналы канала 20, кп 2 используется следующая команда:

```
{"query": "tms-ts", "ch": 20, "rtu": 2}
```

Тогда ответ шлюза может выглядеть так:

```
{
  "total":7,
  "lastId":7,
  "items":[
    {"n":"Демо ОВ 220 кВ В-220","id":1},
```



```
{
  "n": "Демо ОВ 220 кВ 1ШП-220", "id": 2},
  "n": "Демо ОВ 220 кВ 2ШП-220", "id": 3},
  "n": "Демо ОВ 220 кВ ОП-220", "id": 4},
  "n": "Демо ОВ 220 кВ ЗН 1ШП-220", "id": 5},
  "n": "Демо ОВ 220 кВ ЗН 2ШП-220", "id": 6},
  "n": "Демо ОВ 220 кВ ЗН ОП-220", "id": 7}
}
```

Пример запроса и ответа интегральных измерения канал 0, КП 1, начиная с измерения с индексом 200.

```
{"query": "tms-tii", "ch": 0, "rtu": 1, "from": 200}
```

```
{
  "total": 8,
  "lastId": 204,
  "items": [
    {"n": "Энергия А+", "id": 201},
    {"n": "Энергия А-", "id": 202},
    {"n": "Энергия Q+", "id": 203},
    {"n": "Энергия Q-", "id": 204}
  ]
}
```

### 3.6 Команды воздействия на данные

Ряд запросов позволяют выполнять команды, воздействующие на данные сервера: устанавливать состояния сигналов, значения измерений и т.п.

Для обеспечения безопасности данных все команды отличает необходимость авторизации в запросе. В связи с этим ручное выполнение запросов с командами является крайне затруднительным и требует автоматизации на программном уровне.

На стороне шлюза задается текстовый ключ, который представляет собой аналог пароля (задание ключа описано в разделе 2.1 документа). Клиент, осуществляющий запрос, должен знать данный ключ и применить его при создании зашифрованного хэш-кода, который и передается шлюзу при запросе.

Структура запроса команды выглядит следующим образом:

```
{
  "query": "...",
  "...",
  "hash": "<ХэшКод>",
  "timestamp": <ВременнаяМетка>
}
```

где

— Временная метка – Unix-время текущего момента с отличием в том, что в значение включены миллисекунды. Если программно невозможно получить настолько точное значение, следует взять Unix-время с точностью до секунды и добавить произвольные три цифры миллисекунд. Обратите внимание, что при

расчете временной метки не нужно учитывать часовой пояс, следует указывать время UTC.

— Хэш-код – зашифрованная строка, позволяющая авторизовать клиента.

Хэш-код формируется из строки следующего вида:

*!IP-gate@Код@ВременнаяМетка!*

Код, как указано выше, задается в настройках шлюза.

Шифрование выполняется с помощью алгоритма SHA-256 (информацию можно найти в сети Интернет, например, <http://ru.wikipedia.org/wiki/SHA-256>).

При ключе-пароле "abcd1234" и временной метке в момент отправки "1360920815204" должна быть сформирована строка:

*!IP-gate@abcd1234@1360920815204!*

при шифровании которой получится следующий хэш:

*e6a7df550b70670de5cc6e96ba61ed69bbf60f31d7f0184ab498026210dc8aba*

Тогда запрос будет иметь следующий вид:

```
{
  "query": "...",
  "...",
  "hash": "e6a7df550b70670de5cc6e96ba61ed69bbf60f31d7f0184ab498026210dc8aba",
  "timestamp": 1360920815204
}
```

### 3.6.1 Установка состояния сигнала

В запросе установки состояния сигнала используется команда "set-ts", адрес измерения указан в ключе "ts", значение – в ключе "value".

Значение может быть 0 или 1.

Пример корректной установки состояния 1 сигнала с адресом 0.10.100:

```
{
  "query": "set-ts",
  "ts": "0.10.100",
  "value": 1,
  "hash": "c609f4d7bd83b0e50521b408fc689e42c9255168ef58f8919a4417a3c81fbb39",
  "timestamp": 1360918299956
}
```

### 3.6.2 Переключение состояния сигнала

Для переключения текущего состояния сигнала (без необходимости указания нового состояния) следует указать в запросе команду "switch-ts" и адрес сигнала в ключе "ts".

Пример корректного запроса переключения сигнала с адресом 0.100.1:



```
{
  "query": "switch-ts",
  "ts": "0.100.1",
  "hash": "857208c2f5850b2beffe9e98140e9cc993f948c9967515839f4cd8eda0ce3a14",
  "timestamp": 1360917842993
}
```

### 3.6.3 Квитирование сигнала

Квитирование сигнала аналогично переключения состояния, только используется команда "ack-ts".

Пример корректного запроса квитирования сигнала с адресом 1.2.3:

```
{
  "query": "ack-ts",
  "ts": "1.2.3",
  "hash": "93c8d05c9a3830fffeb968b7f0e1019180f8840468c3f340fddacc45aa7dd955",
  "timestamp": 1360918227280
}
```

### 3.6.4 Установка значения измерения

Установка измерения аналогична установке сигнала: используется команда "set-ti", адрес измерения указан в ключе "ti", значение – в ключе "value".

Значение обязательно должно быть либо целым числом, либо числом с плавающей точкой (символом разделителя разрядов является точка).

Пример корректного запроса установки значения "-13,7" измерению с адресом 0.2.6:

```
{
  "query": "set-ti",
  "ti": "0.2.6",
  "value": -13.7,
  "hash": "5db6ad8e86f61c551d835377bad358c7c960fd376d3592483c76c3edc083446d",
  "timestamp": 1360918411856
}
```

## Приложение А

### Расшифровки сообщений об ошибках шлюза

Таблица А.1 – сообщения об ошибках

Сообщение	Расшифровка
Gate is closing now	Шлюз находится в состоянии выключения, получив соответствующую команду от администратора комплекса
No data	Шлюз не получил строку запроса
Query too long	Слишком длинная строка запроса
Events not supported by server	Архив событий шлюза не поддерживается сервером
Commands not supported	Команды воздействия на данные не поддерживаются (не задан пароль)
Server is disconnected	У шлюза отсутствует связь с сервером
ERROR: invalid format	Некорректный JSON формат запроса
ERROR: query type missing	Не указан тип запроса
ERROR: query type unknown	Неизвестный тип запроса
ERROR: start time missing or invalid	Начальное время запроса архива отсутствует или является некорректным
ERROR: end time missing or invalid	Конечное время запроса архива отсутствует или является некорректным
ERROR: invalid start-end times	Некорректность времен запроса архива, одна из причин – время конца архива раньше времени начала
ERROR: step missing or invalid	Шаг выборки архива отсутствует или является некорректным
ERROR: retrospective is too large	Слишком большой архив измерений, требуется сократить выборку
ERROR: too many TS in query	Слишком много сигналов в одном запросе, требуется сократить запрос
ERROR: too many TI in query	Слишком много измерений в одном запросе, требуется сократить запрос
ERROR: too many TII in query	Слишком много интегральных измерений в одном запросе, требуется сократить запрос
ERROR: events quantity missing or invalid	Число запрашиваемых событий отсутствует или является некорректным
ERROR: event ID is invalid	Номер запрашиваемого события отсутствует или является некорректным
ERROR: invalid importance filter	Некорректный фильтр важности событий
ERROR: invalid type filter	Некорректный фильтр типов событий
ERROR: invalid address filter	Некорректный адрес сигнала для архива событий
ERROR: events archive is too large	Слишком большой архив событий, требуется сократить выборку
ERROR: channel ID is missing or invalid	Номер канала при обращении к базе данных телеметрии отсутствует или является некорректным



## Продолжение таблицы А.1

ERROR: RTU ID is missing or invalid	Номер КП при обращении к базе данных телеметрии отсутствует или является некорректным
ERROR: invalid starting point ID	Некорректный стартовый номер объекта при обращении к базе данных телеметрии
ERROR: TS missing or invalid	Адреса сигнала внутри команды отсутствует или является некорректным
ERROR: TI missing or invalid	Адреса измерения внутри команды отсутствует или является некорректным
ERROR: command failure	Ошибка при выполнении команды: изменение данных не зафиксировано
ERROR: command event was added	Команда успешно изменила данные, но не попала в журнал событий
ERROR: hash missing or invalid	Хэш команды отсутствует или является некорректным
ERROR: timestamp missing or invalid	Временная метка команды отсутствует или является некорректной
ERROR: incorrect hash	Неверный хэш при выполнении команды: либо он сформирован неверно, либо пользователь ошибся при вводе пароля