

# Tango module for WebSocket connection

Elkin Vladimir.

Joint Institute for Nuclear Research.

Dubna, Russia

# Протокол WebSocket

- WebSocket – (стандарт [RFC 6455](#)) это протокол полнодуплексной связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени. (Wiki)
- Протокол работает над HTTP(S)
- Передача данных текстового формата (разрешенная кодировка UTF-8)
- Передача бинарных данных
- Поддерживается всеми современными браузерами
- Определено две URI схемы. ws – для нешифрованного соединения, wss – для зашифрованного соединения.

# Установка

## Соединение

```
1 GET /chat HTTP/1.1
2 Host: server.example.com
3 Upgrade: websocket
4 Connection: Upgrade
5 Origin: http://javascript.ru
6 Sec-WebSocket-Key: Iv8io/9s+lYFgZWcXczP8Q==
7 Sec-WebSocket-Version: 13
```

**GET, Host** - Стандартные HTTP-заголовки из URL запроса

**Upgrade, Connection** - Указывают, что браузер хочет перейти на websocket.

**Origin** - Протокол, домен и порт, откуда отправлен запрос.

**Sec-WebSocket-Key** - Случайный ключ, который генерируется браузером: 16 байт в кодировке Base64.

**Sec-WebSocket-Version** - Версия протокола.  
(<https://learn.javascript.ru/websockets>)

# Установка соединения

Если сервер понимает и разрешает WebSocket соединение, приходит сообщение следующего вида

```
1 HTTP/1.1 101 Switching Protocols
2 Upgrade: websocket
3 Connection: Upgrade
4 Sec-WebSocket-Accept: hsB1buDTkk24srzEOTBU1ZA1C2g=
```

**Sec-WebSocket-Accept** - перекодированный по специальному алгоритму ключ Sec-WebSocket-Key. Браузер использует её для проверки, что ответ предназначается именно ему.

Далее для передачи любой стороной текстового сообщения отправляется дата-фрейм следующего вида **0x00**, **<строка в кодировке UTF-8>**, **0xFF**

# Танго-модуль WebSocketDS

## WebSocketDS (WS-DS)

- Данный танго-модуль используется для связи танго-устройств с внешним миром по протоколу WebSocket. Поддерживается автоматическое получение обновляемых данных, получение данных по запросу, а также выполнение команд.
- В качестве клиентов могут использоваться как браузерные приложения (созданием специального javascript объекта), так и десктопные приложения.
- Для создания десктопных приложений существуют библиотеки для практически всех современных сред. QT, C#, boost
- Для запросов, а также результатов используется формат JSON.
- Есть также возможность возвращать данные типа `Tango::DevVarCharArray` напрямую. Может использоваться для получения бинарных данных, например изображений

# WS-DS - Property

The screenshot shows the Jive 6.9 interface with the following details:

- Window title: Jive 6.9 [127.0.0.1:10000]
- Menu bar: File Edit Tools Filter
- Address bar: Server:/WebSocketDS/testws2/WebSocketDS/test/test/wsfortest2/Properties
- Left pane: Tree view showing a hierarchy of devices and properties. The selected path is: Server > Device > Class > testws2 > WebSocketDS > test/test/wsfortest2 > Properties.
- Right pane: "Device properties [test/test/wsfortest2]" table.

Property name	Value
Attributes	bool_spectr
AuthDS	test/webauth/my1
Commands	AttrDevFloatSpectr;precf=5;pipecomm; ComDevShort ComDevFloat
DeviceServer	test/test/testingwebsock
Options	group;notshrtatt;uselog;mode=cli_all_ro
PipeName	PipeAttr AttrDevFloatSpectr;precf=5
Port	7777
SubDevices	test/test/testingwebsock

Buttons at the bottom: Refresh, Apply, New property, Copy, Delete

# WS-DS - Property

- **Port** — Прослушиваемый порт при соединении; (DevShort)
- **DeviceServer** - tango id используемого устройства. Пример: DOMAIN/FAMILY/MEMBER. В случае использования групп устройств, используется шаблон. Параметром шаблона может быть простое имя устройства или шаблон имени устройства (например, domain\_\*/family/member\_\*); (string)
- **Attributes** — список атрибутов устройства, которые вы хотите считывать, если требуется считывать все атрибуты добавьте `__all_attrs__` (в групповом режиме не действует); (array of string)
- **Commands** — список команд устройства,

# WS-DS - Property

- **PipeName** - PipeName для pipe устройства, также список атрибутов с опциями для вывода; (array of string)
- **AuthDS** — танго сервер, отвечающий за аутентификацию и авторизацию пользователя, при наличии исполняемых команд; (string)
- **Secure** —Использование защищённого wss соединения, иначе; (bool)
- **Certificate** — полный путь к используемому сертификату (if Secure = true); (string)
- **Key** - полный путь к используемому файлу с Private key (if Secure = true); (string)



# WS-DS - Property

- **MaxNumberOfConnections** - максимальное число соединений. Если будет достигнут предел, последующие соединения будут прерваны ошибкой 400 Bad Request. Если задано значение 0, количество соединений не будет ограничено.; (DevUShort)
- **MaximumBufferSize** - максимальный размер буфера для каждого соединения в КиБ. Значение по умолчанию 1000. Возможные значения от 1 до 10000 (если указать значения не входящие в заданный диапазон будет выставлено значение по умолчанию). При превышении заданного максимального размера буфера, соединение будет прервано со стороны сервера; (DevULong)

# WS-DS - Property

- **ResetTimestampDifference** - Разница таймстампов (в секундах) после которого будет произведена перезагрузка WS сервера. Расчёт разницы производится в методе CheckPoll между тайстампом обновляемым в методе UpdateData и текущим таймстампом. Минимальное значение 60.; (DevUShort)
- **Options** - дополнительные опции. Перечень дополнительных опций для девайса в формате opt1;opt2=val;opt3

# WS-DS - Attributes

- **JSON** - текущий JSON-output (Актуален, если используется один из SERVER режимов); (DevString)
- **TimestampDiff** - текущая разница таймстампов (обновляется в методе CheckPoll. Значение Polling по умолчанию для него 10 секунд); (DevULong)
- **NumberOfConnections** - текущее число клиентов (соединений). (DevULong)

# WS-DS - Commands

- **CheckPoll** – Команда для проверки разницы timestamp текущего и считанного при последнем обновлении. При превышении разницы 60 секунд производится жёсткая перегрузка танго-модуля (если используется серверный режим). (тип argin argout DevVoid)
- **Reset** – Команда для жёсткой перезагрузки танго-модуля (тип argin argout DevVoid)
- **UpdateData** – Команда для обновления данных (с атрибутов и pipe прописанного танго-девайса) Актуален, если используется серверный режим (тип argin argout DevVoid)

# WS-DS Property “Options”

Свойство **Options** для танго-девайса имеет формат ***opt1;opt2=val;opt3***, где *opt* - имя опции, *val* - значение, если присутствует. Опции разделяются точкой с запятой , ставить её нужно только между опциями.

Опции применяются для выставления дополнительных параметров работы танго-модуля

# WS-DS Property “Options” перечень опций

- **group** - Без дополнительных значений. Использование групп девайсов. Свойство DeviceServer должно быть задано в соответствующем для групп формате.
- **uselog** - Без дополнительных значений. Использование записи в журнал при выполнении команд.
- **tident** - С дополнительным значением. Используемый тип авторизации. Используются три типа rndid - RANDIDENT, rndid2 - RANDIDENT2 и rndid3 RANDIDENT3, а также smpl - SIMPLE (он же используется по умолчанию)
- **mode** - С дополнительным значением. Используемый режим работы модуля.
- **tm100ms** - Без дополнительных значений. По умолчанию, минимальное значение для периода обновления установленного клиентом должно быть больше, либо равно 1000 мсек. При установке данного параметра в Options минимальное значение будет 100 ms.

# Режимы работы WS-DS

Режим работы выставляется в Property “Options”.  
Формат: **mode=выбранный\_режим**

Всего имеется девять режимов, являющиеся комбинацией перечисленных

- **SERV or/and CLIENT** - При серверном режиме разрешается чтение атрибутов и pipe, а также выполнение команд только для танго-устройства (группы) прописанного в Property “DeviceServer”. В клиентном режиме танго-устройства указываются клиентом.
- **ALL/ALIAS** - Разрешается чтение атрибутов и pipe, а также выполнение команд либо для любых танго-девайсов, либо только для имеющих alias
- **RO/RW** - Разрешается либо только чтение атрибутов и pipe, либо также выполнение команд.

# Клиентный режим.

Если используется любой **CLIENT** режим, клиент может управлять обновляемыми данными. Управляется как значение периода обновления, так и содержание данных (перечень девайсов, атрибутов pipe).

Также возможны единичные выполнения команд, а также чтения данных с атрибутов и pipe



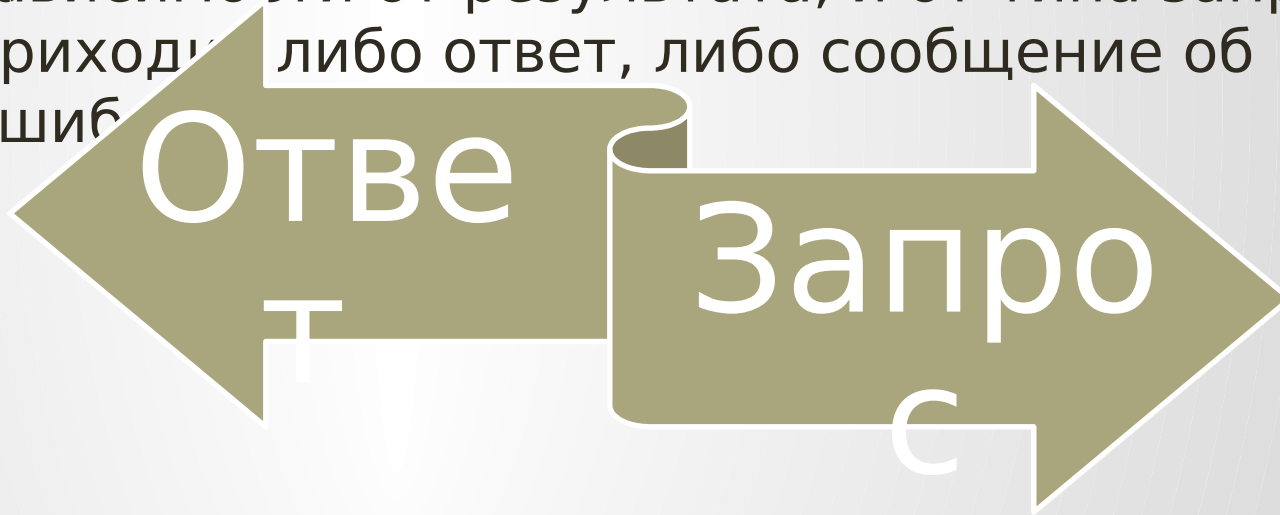
# Клиентный режим.

Перечень команд для клиентного управления обновляемым выводом информации:

- **timer\_start** - Включение таймера. Для включения должны быть также переданы значение таймера в миллисекундах
- **timer\_stop** - Выключение таймера.
- **timer\_change** - Изменения периода обновления данных.
- **timer\_add\_devs** - Добавление девайсов в список прослушиваемых.
- **timer\_remove\_devs** - Удаление девайсов из списка прослушиваемых.
- **timer\_upd\_devs\_add** - Добавление в уже включённые девайсы атрибуты или pipe.
- **timer\_upd\_devs\_rem** - Удаление из уже включённых девайсов атрибуты или pipe.

# Клиентный режим.

Для единичного запроса на выполнение команды или чтения данных с атрибутов или pipe отправляется JSON сообщение. В зависимости от результата, и от типа запроса приходит либо ответ, либо сообщение об ошибке.



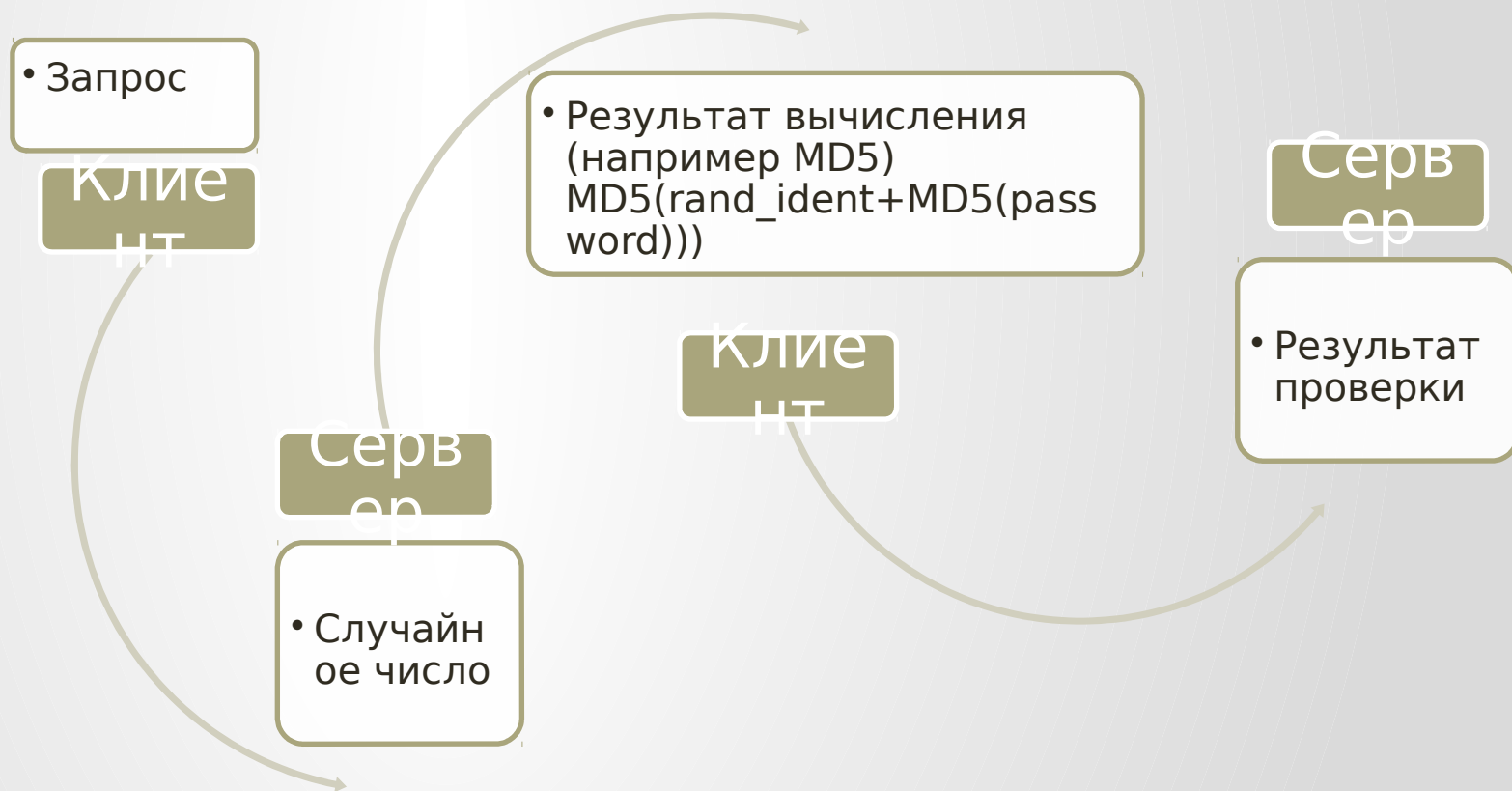
Для идентификации ответа, при каждом запросе следует отправлять id запроса. Это может быть как число, так и строка

# Методы аутентификации

- **SIMPLE** - Самый простой. Производится при подключении. В данном методе логин и пароль должны быть прописаны в URL (пример: ws://ip:port?login=log&password=pass)
- **USERANDIDENT** - Также производится при подключении, но в URL, должны быть прописаны login rand\_ident (случайное число/слово хранящееся/сгенерированное у клиента) rand\_ident\_hash (например MD5(rand\_ident + MD5(password)))
- **USERANDIDENT3** - Принцип тот же что и в **USERANDIDENT**, но аутентификация проводится уже после соединения, путём отправления login rand\_ident и rand\_ident\_hash в JSON сообщении. При успехе, пользователь получает возможность отправления запросов на выполнения команд.

# Методы аутентификации

## Метод USERANDIDENT2



# Методы аутентификации

Аутентификация и авторизация проводится в специальном танго-девайсе, прописанном в Property “AuthDS”.

Для корректного результата в этом девайсе должны содержаться методы **check\_user** (при методе SIMPLE), и **check\_user\_ident** (При методах USERANDIDENT)

Также сервер должен содержать метод **check\_permissions**, в котором производится проверка прав пользователя. В argin передается запрашиваемый танго-девайс, запускаемая команда, IP, с которого производится запуск и логин.

# Журналирование

Также есть возможность записи в журнал результаты запросов на выполнение команд.

Для этого, танго-девайс “AuthDS”, должен также содержать метод **send\_log\_command\_ex** который записывает в таблицу следующие данные:

- **argin[0]** = timestamp\_string (UNIX\_TIMESTAMP)
- **argin[1]** = login
- **argin[2]** = deviceName (либо имя девайса в танго-формате, или шаблон имени девайса)
- **argin[3]** = IP
- **argin[4]** = commandName (имя команды)
- **argin[5]** = commandJson (вводимая команда в json формате)
- **argin[6]** = statusBool (1 если запуск успешен, иначе 0)
- **argin[7]** = isGroup (1 если используется группа, иначе 0)

# Дополнительные параметры

Список доступных на данный момент параметров

- **prefc, precs, prec** - Изменение точности и форматирование выводимых значений атрибутов, pipe и команд типа с плавающей запятой. Для команд, атрибутов и pipe
- **bindata** - Команды с выводом в двоичном формате. Только для команд
- **Niter** - Установление периодичности вывода значений для атрибутов. Для выставления периодичности к имени атрибута в свойстве следует добавить **;niter=N/M** либо **;niter=N**  
Пока только для атрибутов

# Планы

- Добавить возможность подписки на tango EVENT
- Добавить возможность выполнения команд в асинхронном режиме



**Спасибо за  
внимание!**