

Distributed Remote Command Execution functional object message protocol

Request protocol

Json fomate of message:

```
{
  "type": <request_type>,
  "data": "<request_data>",
  "id": <request_id>,
  "subtasks": [<subtasks_array>]
}
```

request_type - numeric integer{0, 1, 2, 3, 4};
0 - Set task to execute, 1 - check task state, 2 - terminate task, 3 - get task's data, 4 - delete task's data, 100 - task state change notification, 101 - resource monitor notification. *Notification message sent by DRCE FO directly in to DTM service without request and not requires response.

request_data - json string data different for each request type, always base64encoded

request_id - request Id treated as unique task Id, numeric (CRC32 or another value).

subtasks_array - array of items the same as "request protocol".

Set task to execute request

Json fomate of "request_data" field:

```
{
  "session": {<session_fields_array>},
  "command": "<command_line>",
  "input": "<input_stream>",
  "files": [<files_array_item>],
  "limits": {<limits_fields_array>}
}
```

session_fields_array - array contains session field items.

command_line - command line that will be executed on remote hosts

input_stream - string of input stream for command that will be copied in to the stdin of the process. Expected utf8 string.

files_array - array of file items.

limits_fields_array - array contains limits for system resources used to check state before run commsnd.

Session fields array:

```
{
  "type": <type>,
  "port": <port>,
  "user": "<user_name>",
  "password": "<user_password>",
  "shell": "<shell_name>",
  "environment": {<environment_array>},
  "home_dir": "<home_directory>",
  "timeout": <timeout>,
}
```

```

    "tmode": <tmode>,
    "time_max": <time_max>,
    "cleanup": <cleanup>
}

```

type - type of remote execution {0 - host shell, 1 - ssh}.

port - tcp port used for session.

user_name - user name for session authorization.

user_password - password for session authorization.

shell_name - name of the shell executor script. If empty or omitted - default shell executor script used from <home_dir>/usr/bin/hce-starter-bash.sh.

environment_array - array of name-value items of OS environment variables that will be exported before command will be executed. Array format:

```

{
  "name": "value", ...
}

```

home_directory - home directory for shell process spawned. If empty - the common node **home_dir** of DRCE FO will be used.

timeout - command execution timeout value, sec, 0 - means not limited.

tmode - threading mode. Values: 1 - synchronous task execution in the same handler's thread, 2 - asynchronous tasks execution, per thread and shell process for each task.

time_max - max execution time, msec. In case of exceeding process terminated by sequential usage of terminate algorithms and correspondent error code and message set for response data. Value need to be aligned on tasks_queue_dump_period (10 sec by default). If value less than tasks_queue_dump_period the task can be executed successfully. Zero value means unlimited time.

cleanup - delete task's data on HCE host. 0 - not delete, 1 - delete; Default is 0 (if empty or omitted).

Files array item:

```

{
  "name": "<file_name>",
  "data": "<file_data>",
  "action": <action_type_mask>
}

```

file_name - name of file that will be processed on host file system before command will be executed. Can contain full or partial path. If path will not specified - current shell.

file_data - data stream that can be written to the file, base64encoded in case of action_type_mask higher bit is set ON.

action_type_mask - action that will be done with file:

Bits meaning:

0 - create before command execution and write data stream;

1 - delete before command execution;

2 - read data stream after execution and return in result response;

3 - delete after command execution;

...

31 - encode/decode content with base64 algorithm.

Limits fields array:

```
{
  "proc_max": <proc_max>,
  "threads_max": <threads_max>,
  "cpu_max": <cpu_max>,
  "iowait_max": <iowait_max>,
  "vram_free": <vram_free>,
  "vram_free_min": <vram_free_min>,
  "rram_free": <rram_free>,
  "rram_free_min": <rram_free_min>,
  "disk_free": <disk_free>,
  "disk_free_min": <disk_free_min>,
  "iowait_usage_max": <iowait_usage_max>,
  "vram_usage_max": <vram_usage_max>,
  "rram_usage_max": <rram_usage_max>,
  "disk_usage_max": <disk_usage_max>,
  "disk_usage_max_bytes": <disk_usage_max_bytes>
}
```

proc_max - max number of processes, numeric
threads_max - max number of threads, numeric
cpu_max - max value of CPU load, %.
iowait_max - max value of iowait load, %.
vram_free - min size of resource RAM, %
rram_free_min - min size of resource RAM bytes, numeric
vram_free_min - min size of virtual RAM, %
vram_free_min - min size of virtual RAM byte, numeric
disk_free - min size of disk space on the execution partition, %
disk_free_min - min size of disk space byte, numeric
iowait_usage_max - max allowed value of common iowait load during execution, %.
vram_usage_max - max allowed value of vram process usage during execution, %.
rram_usage_max - max allowed value of rram process usage during execution, %.
disk_usage_max - max allowed value of disk process usage during execution, %.
disk_usage_max_bytes - max allowed value of disk process usage during execution, bytes.

Check task request

Json format of "request_data" field:

```
{
  "type": <check_type>,
}
```

check_type - numeric, 1 - return simple status information, 2 - return extended status information (complete status file data).

Terminate task request

Json format of "request_data" field:

```
{
  "alg": <algorithm_type>,
  "delay": <delay_value>,
  "repeat": <repeat_value>,
  "signal": <signal_value>,
  "cleanup": <cleanup>
}
```

algorithm_type - numeric, 1 - default algorithm: send SIG_TERM signal, wait **delay_value** ms and send SIG_KILL signal, repeat **repeat_value** times with the SIG_KILL signal, 2 - custom signal algorithm, the same as 1, but uses only **signal_value** signal).

delay_value - delay before next try, numeric, msec must be greater than zero.

repeat_value - tries repeat number, numeric, must be greater than zero.

signal_value - Unix/Linux signal number, see signal(7), numeric.

cleanup - delete task's data on HCE host. 0 - not delete, 1 - delete; Default is 0 (if empty or omitted).

Get task's data request

Json format of "request_data" field:

```
{
  "type": <fetch_type>,
}
```

fetch_type - numeric, 1 - delete data after fetch, 2 - not delete data after fetch.

Delete task's data request

Json format of "request_data" field:

```
{}
```

Response protocol

[<result_item>, ...]

result_item format:

```
{
  "error_code": <error_code>,
  "error_message": "<error_message>",
  "id": "<request_id>",
  "type": "<request_type>",
  "host": "<node_host>",
  "port": "<node_port>",
  "state": "<state>",
  "pid": "<pid>",
  "stdout": "<stdout_stream>",
  "stderr": "<stderr_stream>",
  "exit_status": "<exit_status>",
  "files": [<files_array>],
  "node": "<node_name>",
  "time": "<time>",
  "fields": <fields_dictionary>,
  "subtasks": [<subtasks_array>]
}
```

error_code - error code of this task request, zero means success.

error_message - error message text, empty if success.

request_id - the same as "request_id" in request query message

request_type - the same as "request_type" in request query message for response on request and ???

state - task execution state, numeric values 0 - finished, 1 - in progress, 2 - set as new, 3 - not found, 4 - terminated before normal finish, 5 - crashed, 6 - not set as new cause to limits of system resources (CPU, RAM, Disk, threads, processes, etc) or input request json error or transport system error, 7 - undefined, 8 - queued to run, 9 - task was successfully deleted, 10 - task is busy (means state can't be received this time and operation can be repeated since short period of time).

pid - process Id.

node_host - node admin host from DRCE config.

node_port - node admin port from DRCE config.

stdout_stream - command stdout stream, base64encoded

stderr_stream - command stderr stream, base64encoded

exit_status - the Linux shell EXITSTATUS value

files_array - array of output files, the same fields set and actions as for request protocol.

node_name - node name string.

time - total task execution time - if it is finished, terminated or crashed and NOW()-START_TIME - if the task in progress.

fields_dictionary - dictionary of fields items:

"name": "value", for example, this fields can be set for "CHECK STATE" or update status callback:

- **TASK_CPU** – task CPU LA,% = 100%-idle
- **TASK_VRAM** – task VRAM usage, Bytes
- **TASK_RRAM** – task RRAM usage, Bytes
- **TASK_THREADS** – task threads number
- **HOST_CPU** – host CPU usage value,%
- **HOST_IOWAIT** – host iowait load, %
- **HOST_VRAM** – host virtual memory usage, %
- **HOST_RRAM** – host rss memory usage, %
- **HOST_DISK** – host hard disk drive usage, %
- **HOST_PROCESSES** – host processes usage, %
- **HOST_THREADS** – host threads usage, %

subtasks_array – array of responses items for subtasks.

*Default value for numeric fields is zero - 0, for string is empty string – "".

Stat variables structure

use for get resource statistics as **NODE_MSG_PROPERTIES** admin command

```
{  
  <timeAsyncTask> : "Stat variable data",  
  <timeSyncTask> : "Stat variable data",  
  <sizeInputBufferSyncTasks> : "Stat variable data",  
  <sizeOutputBufferSyncTasks> : "Stat variable data",  
  <sizeInputBufferAsyncTasks> : "Stat variable data",  
  <sizeOutputBufferAsyncTasks> : "Stat variable data",  
  <cpuUsageAsyncTasks> : "Stat variable data",  
  <memoryUsageAsyncTasks> : "Stat variable data",  
  <countSyncTasksForMinute> : "Stat variable data",  
  <countAsyncTasksForMinute> : "Stat variable data",  
  <countSyncTasks> : numeric,  
  <countAsyncTasks> : numeric,  
  <countSyncTasksFail> : numeric,  
  <countAsyncTasksFail> : numeric  
}
```

timeAsyncTask - time working of async task,

timeSyncTask - time working of sync task,

sizeInputBufferSyncTasks – size of input buffer stdin for sync tasks,

sizeOutputBufferSyncTasks – size of output buffer stdout for sync tasks,
sizeInputBufferAsyncTasks - size of input buffer stdin for async tasks,
sizeOutputBufferAsyncTasks - size of output buffer stdout for async tasks,
cpuUsageAsyncTasks – cpu usage only async tasks,
memoryUsageAsyncTasks – memory usage only async tasks,
countSyncTasksForMinute – count of sync tasks for minute,
countAsyncTasksForMinute - count of async tasks for minute,
countSyncTasks – full count sync tasks for all time,
countAsyncTasks – full count async tasks for all time,
countSyncTasksFail – full count sync tasks returned error for all time,
countAsyncTasksFail – full count async tasks returned error for all time

Stat variable data structure

{ <min>, <avg>, <max> }

min – minimum value,

avg – average value,

max – maximum value.

All values have numeric types and zero as default value.

Task status data structure

used for **get current tasks** as **DRCE_GET_TASKS** admin command

{ <id>, <sdate>, <state>, <tmode> }

id – unique task ID,

sdate – timestamp of start task ,

state -- task execution state, numeric values 0 – finished, 1 – in progress, 2 – set as new, 3 – not found, 4 – terminated before normal finish, 5 – crashed, 6 – not set as new cause to limits of system resources (CPU, RAM, Disk, threads, processes, etc) or input request json error, 7 – undefined, 8 – queued to run, 9 – task was successfully deleted, 10 – task is busy.

tmode – threading mode. Values: 1 – synchronous task execution in the same handler's thread, 2 – asynchronous tasks execution, per thread and shell process for each task.

***Task status data structure** used for **task log file** and for get current tasks queue.

Get current tasks queue content serialized json string of array:

[<Task status data structure>, NULL]

Task info data structure

used as encapsulation **list info task** for **DRCE_GET_TASKS_INFO** admin command

{ <id>, <info>, <modified> }

id – unique task ID,

info – content task's info file,

modified – timestamp of last modified

List all tasks info structures use for **get list info** about all tasks as **DRCE_GET_TASKS_INFO** admin command

Get list all tasks content serialized json string of array:

[<Task info data structure>, NULL]

Order of resource load from data node

used for **balance message transport**

It is string use ',' as delimiter use for calculate load balancing on manager node.

“<cpu>,<iowait>,<vram>,<rram>,<disk>,<processes>,<threads>,<load_average>”

<cpu> – CPU usage value %,

<iowait> - iowait load %,
<vram> - virtual RAM usage %,
<rram> - resource RAM usage %,
<disk> - hard disk usage %,
<processes> - processes usage from max value %,
<threads> - threads usage from max value %,
<load_average> - load average value %

Resource usage data structure protocol

used for get resource usage as `NODE_RESOURCE_USAGE` admin command

```
{  
    "CPU": <cpu>,  
    "IOWAIT": <iowait>,  
    "VRAM": <vram>,  
    "RRAM": <rram>,  
    "DISK": <disk>,  
    "PROCESSES": <processes>,  
    "THREADS": <threads>,  
    "LOADAVG": <load_average>  
}
```

<cpu> - CPU usage value %,
<iowait> - iowait load %,
<vram> - virtual RAM usage %,
<rram> - resource RAM usage %,
<disk> - hard disk usage %,
<processes> - processes usage from max value %,
<threads> - threads usage from max value %,

<load_average> - load average value %

Appendix 1.

Error codes and explanation

Code	Description
0	No errors - return successful without errors
1	Parse error – wrong json format or another parse error.
2	General internal error of instantiation functional object
3	General internal error of make default JSON.
4	General internal error of set home directory.
5	General internal error of set mandatory environment variables
11	General internal error of serialization.
12	General internal error of unserialization.
13	General internal error unserialization json with bad format
14	General internal error task ID has not allowed range
15	General internal error shell script not exist
16	General internal error shell script cannot execute
1xx	General FO errors
100	General internal FO error
101	General command execution error
102	General error set environments
103	General error bad input params
104	General error of create new task
105	General error bad type of task request
106	General error create some object
107	General error appeared during execute task
108	General error read status file

- 109 General error read data file
- 110 General error read request file
- 111 General error get data, but task continues to run
- 112 General error read bad data content
- 113 General error terminate task
- 114 General error invalid pid process
- 115 General error execute notification functor
- 116 General error terminated expired task
- 117 General error terminated subtask after expired main task
- 118 General error canceled task
- 119 General error exceeded resource limits
- 120 General error cleanup task data
- 121 General error delete task data
- 122 General error build list all tasks on cluster node
- 123 General error getting check state, because task is busy

- 2xx FS operations error
- 201 Can't create target source file before execute
- 202 Can't delete target source file before execute
- 203 Can't read target source file after execute
- 204 Can't delete target source file after execute

- 5xxx Cluster node errors
- 5000 No target node clients available
- 5001 Node mode not supported

Appendix 2. Examples of json queries

Example 1

```
//Request protocol
//Request example set task to execute request
{"data": "eyJzZXNzaW9uljp7InR5cGUiOiJAsInRtb2RlljoyLCJ0aW1lX21heCI6NDAwMDAsInBvcnQi\r\nOjAsInVzZXliOiIiLCJwYXNzd29yZCI6IiIsInNoZWxsljoiIiwiaW52aXJvbm1lbnQiOiI7\r\nIklxJTkVTlJoiODAiLCJIT1NUTkFNRSi6Im5vZGUyMiJ9XX0sImNvbW1hbmQiOiJzb3J0IiwiaW52aXJvbm1lbnQiOiI7\r\nnaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4iLCJmaWxlcyl6W3sibmFtZSI6ImZpbGUxliwiZGF0\r\nYSi6InRlc3QgY29udGVudCBvZiBmaWxIMVxuliwiYWNOaW9uljoxM30seyJuYW1lIjoiZmls\r\nZTIiLCJkYXRhIjoidGVzdCBjb250ZW50IG9mIGZpbGUyXG4iLCJhY3Rpb24iOiJF9XX0=","id":567,"type":0}
// json encoded Base64 in field 'data'
{
  "session": {
    "type": 0,
    "tmode": 2,
    "time_max": 40000,
    "port": 0,
    "user": "",
    "password": "",
    "shell": "",
    "environment": [
      {
        "LINES": "80",
        "HOSTNAME": "node22"
      }
    ]
  },
  "command": "sort",
  "input": "def\nabc\n123\n",
  "files": [
    {
      "name": "file1",

```

```
        "data": "test content of file1\n",
        "action": 13
    },
    {
        "name": "file2",
        "data": "test content of file2\n",
        "action": 1
    }
]
}
//Response protocol
//Response example set task to execute request
[
    {
        "error_code": 0,
        "error_message": "",
        "exit_status": 0,
        "fields": [],
        "files": [],
        "host": "NODE_HOST",
        "id": 567,
        "node": "NODE_NAME",
        "pid": 0,
        "port": "NODE_PORT",
        "state": 2,
        "stderr": "",
        "stdout": "",
        "subtasks": [],
        "time": 0,
        "type": 0
    }
]
```

Example 2

```
//Request protocol
//Request example check task state request
{"data":"eyJ0eXBlljoyfQ==","id":567,"type":1}

// json encoded Base64 in field 'data'
{"type":2}

//Response protocol
//Response example check task state request
[{"error_code":0,"error_message":"","exit_status":0,"fields":{"HOST_CPU":"55.120000","HOST_DISK":"4.220000","HOST_IOWAIT":"0.000000","HOST_PROCESSES":"0.000450","HOST_RRAM":"2201828045.250000","HOST_THREADS":"0.510000","HOST_VRAM":"0.520000","TASK_CPU":"0.000000","TASK_RRAM":"2121728","TASK_THREADS":"2","TASK_VRAM":"16707584"},"files":[{"action":13,"data":"test content of file1\n","name":"file1"}],"host":"NODE_HOST","id":567,"node":"NODE_NAME","pid":2450,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTIzCmFiYwpkZWYK","time":605,"subtasks":[],"type":0}]
```

Example 3

```
//Request protocol
//Request example terminate task request
{"data":"eyJhbGciOiJIc2UiLCJ0eXBlIjoiYm9keSIsImVudCIsImVudC1kaXIiOiJ0eXBlIjoiYm9keSIsImVudC1kaXIiOiJ0eXBlIjoiYm9keSIsImVudC1kaXIiOiJ0eXBlIjoiYm9keS","id":568,"type":2}

// json encoded Base64 in field 'data'
{"alg":2,"cleanup":0,"delay":1000,"repeat":3,"signal":9}

//Response protocol
//Response example terminate task request
[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":568,"node":"NODE_NAME","pid":4498,"port":"NODE_PORT","state":4,"stderr":"","stdout":"","time":8249,"subtasks":[],"type":2}]
```

Example 4

```
//Request protocol
//Request example get task's data request
{"data":"eyJ0eXBlljoyfQ==","id":567,"type":3}

// json encoded Base64 in field 'data'
{"type":2}

//Response protocol
//Response example get task's data request

[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[{"action":13,"data":"test content of file1\n","name":"file1"}],"host":"NODE_HOST","id":567,"node":"NODE_NAME","pid":4655,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTIzCmFiYwpkZWYK","time":736,"subtasks":[],"type":0}]
```

Example 5

```
//Request protocol
//Request example delete task's data request
{"data":"e30=","id":567,"type":4}

// json encoded Base64 in field 'data'
{}

//Response protocol
//Response example delete task's data request

[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":567,"node":"NODE_NAME","pid":4655,"port":"NODE_PORT","state":9,"stderr":"","stdout":"","subtasks":[],"time":5401,"subtasks":[],"type":4}]
```

Example 6

```
//Request protocol
```

```
//Request example for local shell request (command 'sort')
```

```
{"data": "eyJzZXNzaW9uIjpb7InR5cGUiOjAsInBvcnQiOjAsInVzZXliOiliLCJ0aW1lX21heCl6MCwi\r\nndG1vZGU  
iOjEsInRpbWVfbWF4IjowLCJwYXNzd29yZCI6IiIsInNoZWxsljoiIiwiaW52aXJv\r\nbnm1IbnQiOlt7IkxJTkVtIjoi  
ODAiLCJIT1NUTkFNRSI6Im5vZGUyMiJ9XX0slmNvbW1hbmQi\r\nnOiJzb3J0IiwiaW5wdXQiOiJkZWZcbmFiY  
1xuMTIzXG4iLCJmaWxlcyl6W3sibmFtZSI6ImZp\r\nnbGUxliwiZGF0YSI6InRlc3QgY29udGVudCBvZiBmaWxl  
MVxuliwiYWNoaW9uIjpbM30seyJu\r\nnYW1lIjoiZmlsZTIiLCJkYXRhIjoidGVzdCBjb250ZW50IG9mIGZpbGUy  
XG4iLCJhY3Rpb24i\r\nnOjF9XX0=", "id": 2345, "type": 0}
```

```
// json encoded Base64 in field 'data'
```

```
{"session": {"type": 0, "port": 0, "user": "", "time_max": 0, "tmode": 1, "time_max": 0, "password": "", "shell": "",  
environment": {"LINES": "80", "HOSTNAME": "node22"}}, "command": "sort", "input": "def\nabc\n123\n",  
files": [{"name": "file1", "data": "test content of file1\n", "action": 13}, {"name": "file2", "data": "test content  
of file2\n", "action": 1}]}
```

```
//Response protocol
```

```
//Response example for local shell request (command 'sort')
```

```
[{"error_code": 0, "error_message": "", "exit_status": 0, "fields": [], "files": [{"action": 13, "data": "test content  
of  
file1\n", "name": "file1"}], "host": "NODE_HOST", "id": 2345, "node": "NODE_NAME", "pid": 4998, "port": "NO  
DE_PORT", "state": 0, "stderr": "", "stdout": "MTIzCmFiYwpkZWYK", "time": 5, "subtasks": [], "type": 0}]
```

Example 7

```
//Request protocol
```

```
//Request example for local shell request (command 'ls -la')
```

```
{"type": 0, "data": "eyJzZXNzaW9uIjpb7InR5cGUiOjAsInBvcnQiOjAsInVzZXliOiliLCJwYXNzd29yZCI6IiIsInNoZ  
WxsljoiIiwiaW52aXJvbnm1IbnQiOlt7IkxJTkVtIjoiODAiLCJIT1NUTkFNRSI6Im5vZGUyMiJ9XX0slmNvbW1hbmQi\r\nn  
OiJzb3J0IiwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4iLCJmaWxlcyl6W10slmxbWl0cyI6eyJwcm9j  
X21heCl6MCwidGhyZWZk19tYXgiOjAsImNwdSI6MCwidnJhbV9mcmVlljowLCJ2cmFtX2ZyZWVfbWluljow  
LCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsImRpc2tfZnJlZSI6MCwiZGlza19mcmVlX21pbil6MH19  
", "id": 12345}
```

```
// json encoded Base64 in field 'data'
```

```
{"session": {"type": 0, "port": 0, "user": "", "password": "", "shell": "", "environment": [], "home_dir": "", "timeou  
t": 0, "tmode": 1, "time_max": 60000}, "command": "ls -  
la", "input": "", "files": [], "limits": {"proc_max": 0, "threads_max": 0, "cpu": 0, "vram_free": 0, "vram_free_min":  
0, "rram_free": 0, "rram_free_min": 0, "disk_free": 0, "disk_free_min": 0}}
```



```
//Response protocol
```

```
//Response example for local shell request (command 'ls -la')
```

```
{{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":12345,"node":"NODE_NAME","pid":5111,"port":"NODE_PORT","state":0,"stderr":"","stdout":"dG90YWwgOTAwOApkcnd4ci14ci14IDEyIGFsZXhhbmRlciBhbGV4YW5kZXIglCAgNDA5NiDQr\\nnNCw0LkgMjlgMTU6MzggLgpkcnd4ci14ci14IDEwIGFsZXhhbmRlciBhbGV4YW5kZXIglCAg\\nNDA5NiDQnNCw0LkgMTUgMTQ6MzMgLi4KZHJ3eHlteHlIteCAgMiBhbGV4YW5kZXIglYWxleGFu\\nZGVyIcAgIDQwOTYgOjZQsNC5IDlxIDE0OjAxlGJhc2U2NF9zcmMKLXJ3eHlteHlIteCAgMSBh\\n\\nbGV4YW5kZXIglYWxleGFuZGVyIcAgICAxNTcg0JDQv9GAIDMwIDExOjU5IGNsZWFLnNoCmRy\\n\\nd3hyLXhyLXggIDlgYWxleGFuZGVyIGFsZXhhbmRlciAgICA0MDk2INCC0LDQuSAyMiAxNToy\\n\\nOCBkYXRhCi1yd3hyLXhyLXggIDlgYWxleGFuZGVyIGFsZXhhbmRlciAgICA0MzZlM2NzI4INCC0LDQr\\nuSAyMiAxNTozOCBEUkNFcmRyd3hyLXhyLXggIDlgYWxleGFuZGVyIGFsZXhhbmRlciAgICA0\\n\\nMDk2INCC0LDQuSAyMiAxNToyMiBLEGVjX3NyYwotcncctci0tci0tIcAxIGFsZXhhbmRlciBh\\n\\nbGV4YW5kZXIglCAgICAgYMiDQnNCw0LkgMjlgMTU6MzUgZmlsZTIKLXJ3LXItLXItLSAgMSBh\\n\\nbGV4YW5kZXIglYWxleGFuZGVyIcAgIDe2NzcgOjZQsNC5IDlxIDE0Jm4IG1ha2VmaWxlcmRy\\n\\nd3hyLXhyLXggIDlgYWxleGFuZGVyIGFsZXhhbmRlciAgICA0MDk2INCC0LDQuSAyMSAxNDow\\n\\nMSBtZXNzYWdlU3RyZWFTX3NyYwotcncctci0tci0tIcAxIGFsZXhhbmRlciBhbGV4YW5kZXIglCAgIDYxNiDQnNCw0LkgMjlg\\n\\nMTU6Mzggc291cmNlcy5tawpkcnd4ci14ci14ICAyIGFsZXhhbmRlciBhbGV4YW5kZXIglCAg\\n\\nNDA5NiDQnNCw0LkgMjlgMTU6MTQgcmRyd3hyLXhyLXggIDlgYWxleGFuZGVyIGFsZXhh\\n\\nr\\n\\nbmRlciAgICA0MDk2INCC0LDQuSAyMiAxNTozOCBzdGF0dXMKLXJ3LXItLXItLSAgMSBhbGV4\\n\\nYW5kZXIglYWxleGFuZGVyIcAgICA0MTUgOjZQsNC5IDlxIDE0Jm4IHRIc3QuY2xhc3MKZHJ3\\n\\neHlteHlIteCAgMiBhbGV4YW5kZXIglYWxleGFuZGVyIcAgIDQwOTYgOjZQsNC5IDlxIDE0Jm4\\n\\nIHRlc3RzCi1ydy1yLS1yLS0gIDEgYWxleGFuZGVyIcAgIDQwOTYgOjZQsNC5IDlxIDE0Jm4IHRp\\n\\nbWVEZWxeS5sb2cKZHJ3eHlteHlIteCAgMyBhbGV4YW5kZXIglYWxleGFuZGVyIcAgIDQwOTYg\\n\\n0JDQv9GAIDE3IDEyOjE3IHVzcgpkcnd4ci14ci14ICAyIGFsZXhhbmRlciBhbGV4YW5kZXIgl\\n\\n\\nCAGNDA5NiDQnNCw0LkgMjlgMTQ6MDAgeG1sQ2xlYW5lcml9zcmMK","time":104,"subtasks":[],"type":"0"}}
```

Example 8

```
//Request protocol
```

```
//Request example for local shell request (command 'ps')
```

```
{"data":"eyJzZXNzaW9uIjpw7lnR5cGUiOjAsInBvcnQiOjAsInVzZXIiOiIlLCJ0aWw1IX21heCl6MCIw\\n\\ndG1vZGUiOjEsInBhc3N3b3Jkljoiiliwic2hIbGwiOiIlLCJlbnp3cm9ubWVudC16W3siTEIO\\n\\nRVMIoiI4MCIslkhPU1ROQU1Fljoibm9kZTlYln1dfSwiY29tbWFLuZCI6InBzID4gX3BzX3Jl\\n\\nc3VsdHMudHh0liwiaW5wdXQiOiIlLCJmaWxlcyI6W3sibmFtZSI6Il9wc19yZXN1bHRzLnR4\\n\\ndC1sImRhGEiOiIlLCJhY3Rpb24iOjEYfV19","id":2345,"type":"0"}}
```

```
// json encoded Base64 in field 'data'
{"session":{"type":0,"port":0,"user":"","time_max":0,"tmode":1,"password":"","shell":"","environment":
[{"LINES":"80","HOSTNAME":"node22"}]},"command":"ps >
_ps_results.txt","input":"","files":[{"name":"_ps_results.txt","data":"","action":12}]}
```

//Response protocol

//Response example for local shell request (command 'ps')

```
[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[{"action":12,"data":" PID TTY
TIME CMD\n 4256 pts/0  00:00:00 bash\n 5273 pts/0  00:00:00 DRCE\n 5278 pts/0  00:00:00 hce-
starter-bas\n 5279 pts/0  00:00:00
ps\n","name":"_ps_results.txt"}],"host":"NODE_HOST","id":2345,"node":"NODE_NAME","pid":5278,"po
rt":"NODE_PORT","state":0,"stderr":"","stdout":"","time":13,"subtasks":[],"type":0}]}
```

Example 9

//Request protocol

//Request example for local shell request (command 'php test.php > result.txt')

```
{"data":"eyJzZXNzaW9uIjpb7InR5cGUiOjAsInBvcnQiOjAsInVzZXliOiliLCJ0aW1lX21heCl6MCwi\r\nndG1vZGU
iOjEsnBhc3N3b3Jkljoiliwic2hIbGwiOiliLCJlbnZpcm9ubWVudCl6W3siTElO\r\nRVMiOi4MClslkhPU1ROQU
1Fljoibm9kZTIyIn1dfSwiY29tbWFuZCI6InBocCB0ZXN0LnBo\r\ncCA+IHJlc3VsdC50eHQiLCJpbmB1dCl6liIsI
mZpbGVzIjpbeyJuYW1lIjojdGVzdC5waHAi\r\nLCJkYXRhIjojPD9waHBcbmVjaG8gJ0hlbGxvLCB3b3JsZCEnO
1xuPz5cbilslmFjdGlvbil6\r\nOX0sIHsibmFtZSI6InJlc3VsdC50eHQiLCJkYXRhIjoiliwiYWw0aW9uIjoxMn1dfQ
==","id":2345,"type":0}
```

// json encoded Base64 in field 'data'

```
{"session":{"type":0,"port":0,"user":"","time_max":0,"tmode":1,"password":"","shell":"","environment":
[{"LINES":"80","HOSTNAME":"node22"}]},"command":"php test.php >
result.txt","input":"","files":[{"name":"test.php","data":"<?php\nnecho 'Hello,
world!'\n?>\n","action":9}, {"name":"result.txt","data":"","action":12}]}
```

//Response protocol

//Response example for local shell request (command 'php test.php > result.txt')

```
[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[{"action":12,"data":"Hello,
world!","name":"result.txt"}],"host":"NODE_HOST","id":2345,"node":"NODE_NAME","pid":5440,"port":
"NODE_PORT","state":0,"stderr":"","stdout":"","time":12,"subtasks":[],"type":0}]}
```

Example 10

```
//Request protocol
```

```
//Request example set async task to execute request use async subtasks
```

```
{ "data": "eyJjb21tYW5kljoic29ydClmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljlwLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljpw7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 500, "subtasks": { "data": "eyJjb21tYW5kljoic29ydClmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\n\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljlwLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljpw7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 501, "subtasks": { "data": "eyJjb21tYW5kljoic29ydClmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\n\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljlwLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljpw7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 503, "subtasks": [], "type": 0, {"data": "eyJjb21tYW5kljoic29ydClmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\n\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljlwLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljpw7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 504, "subtasks": [], "type": 0, {"data": "eyJjb21tYW5kljoic29ydClmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\n\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljlwLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljpw7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 502, "subtasks": { "data": "eyJjb21tYW5kljoic29ydClmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\n\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljlwLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljpw7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 505, "subtasks": [], "type": 0, {"data": "eyJjb21tYW5kljoic29ydClmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\n\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4i
```

```
OjAsInRocmVhZHNfbWF4\r\nljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uIjpw7ImNsZWFu\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nnliInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=","id":506,"subtasks":[],"type":0},"type":0},"type":0}
```

/Response protocol

//Response example set async task to execute request use async subtasks

```
{{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":500,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":2,"stderr":"","stdout":"","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":501,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":503,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0},{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":504,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0}],{"time":0,"type":0},{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":502,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":505,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0},{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":506,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0}],{"time":0,"type":0},"time":0,"type":0}}
```

Example 11

//Request protocol

//Request example set async task to execute request use sync subtasks

```
{"data":"eyJjb21tYW5kljoic29ydCIsImZpbGVzIjpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nnLCJsaW1pdHMiOnsiY3B1IjowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\nnb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\nnIjowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uIjpw7ImNsZWFu\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nnliInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=","id":500,"subtasks":[{"data":"eyJjb21tYW5kljoic29ydCIsImZpbGVzIjpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nnLCJsaW1pdHMiOnsiY3B1IjowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\nnb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\nnIjowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uIjpw7ImNsZWFu\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nnliInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\nndG1vZGUiOjEsInR5cGUiOjAsInVzZXliOiliifX0=","id":501,"subtasks":[{"data":"eyJjb21tYW5kljoic29ydCIsImZpbGVzIjpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nnLCJsaW1pdHMiOnsiY3B1IjowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\nnb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\nnIjowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJz
```

```
XNzaW9uljp7ImNsZWFu\r\ndXAiOjAsImVudmlyb25tZW50ljbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\ndG1vZGUiOjEslR5cGUiOjAsInVzZXliOiliifX0=","id":503,"subtasks":[],"type":0},{ "data": "eyJjb21tYW5kljoic29ydClslmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1lJowLCJkaXNrX2ZyZWUiOjAsImRpc2tZnJlZV9taW4iOjAsInBy\r\nnb2NfbWF4lJowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljp7ImNsZWFu\r\ndXAiOjAsImVudmlyb25tZW50ljbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\ndG1vZGUiOjEslR5cGUiOjAsInVzZXliOiliifX0=","id":504,"subtasks":[],"type":0},{ "data": "eyJjb21tYW5kljoic29ydClslmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1lJowLCJkaXNrX2ZyZWUiOjAsImRpc2tZnJlZV9taW4iOjAsInBy\r\nnb2NfbWF4lJowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljp7ImNsZWFu\r\ndXAiOjAsImVudmlyb25tZW50ljbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\ndG1vZGUiOjEslR5cGUiOjAsInVzZXliOiliifX0=","id":502,"subtasks":{ "data": "eyJjb21tYW5kljoic29ydClslmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1lJowLCJkaXNrX2ZyZWUiOjAsImRpc2tZnJlZV9taW4iOjAsInBy\r\nnb2NfbWF4lJowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljp7ImNsZWFu\r\ndXAiOjAsImVudmlyb25tZW50ljbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\ndG1vZGUiOjEslR5cGUiOjAsInVzZXliOiliifX0=","id":505,"subtasks":{ "data": "eyJjb21tYW5kljoic29ydClslmZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1lJowLCJkaXNrX2ZyZWUiOjAsImRpc2tZnJlZV9taW4iOjAsInBy\r\nnb2NfbWF4lJowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljp7ImNsZWFu\r\ndXAiOjAsImVudmlyb25tZW50ljbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\ndG1vZGUiOjEslR5cGUiOjAsInVzZXliOiliifX0=","id":506,"subtasks":[],"type":0},"type":0},"type":0}
```

/Response protocol

//Response example set async task to execute request use sync subtasks

```
{ "error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":500,"node":"NODE_NAME","pid":5119,"port":"NODE_PORT","state":1,"stderr":"","stdout":"","subtasks":{ "error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":501,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":{ "error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":503,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0},{ "error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":504,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0},{ "error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":502,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":{ "error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":505,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0},{ "error_code":0,"error_message":"","exit_status":0,"fields":
```

```
s":[], "files":[], "host": "NODE_HOST", "id": 506, "node": "NODE_NAME", "pid": 0, "port": "NODE_PORT", "state": 8, "stderr": "", "stdout": "", "subtasks": [], "time": 0, "type": 0}, {"time": 0, "type": 0}, {"time": 37, "type": 0}]
```

Example 12

//Request protocol

//Request example set sync task to execute request use async subtasks

```
{"data": "eyJjb21tYW5kljoic29ydCIsImZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJyZmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9ulj7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliilsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 500, "subtasks": [{"data": "eyJjb21tYW5kljoic29ydCIsImZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJyZmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9ulj7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliilsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 501, "subtasks": [{"data": "eyJjb21tYW5kljoic29ydCIsImZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJyZmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9ulj7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliilsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 502, "subtasks": [{"data": "eyJjb21tYW5kljoic29ydCIsImZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJyZmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9ulj7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliilsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 503, "subtasks": [{"data": "eyJjb21tYW5kljoic29ydCIsImZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJyZmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9ulj7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliilsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 504, "subtasks": [{"data": "eyJjb21tYW5kljoic29ydCIsImZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJyZmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9ulj7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliilsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0=", "id": 505, "subtasks": [{"data": "eyJjb21tYW5kljoic29ydCIsImZpbGVzljpbXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1ljowLCJkaXNrX2ZyZWUiOjAsImRpc2tfnJlZV9taW4iOjAsInBy\r\n\r\nb2NfbWF4IjowLCJyZmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\n\r\nljljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9ulj7ImNsZWFu\r\n\r\nndXAiOjAsImVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\n\r\nr\nliilsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\n\r\nndG1vZGUiOjIsInR5cGUiOjAsInVzZXliOiliifX0="}]}
```

```
UiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4i
OjAsInRocmVhZHNfbWF4\r\nljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uIjpw
7ImNsZWFu\r\nndXAiOjAsImVudmlyb25tZW50IjpwXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nnl
iIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\nndG1vZGUiOjIsInR
5cGUiOjAsInVzZXliOiliX0=","id":506,"subtasks":[],"type":0},"type":0},"type":0}
```

/Response protocol

//Response example set sync task to execute request use async subtasks

```
[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":500,"
node":"NODE_NAME","pid":5202,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTIzCmFiYwpc
ZWIYK","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE
_HOST","id":501,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":2,"stderr":"","stdout":"
","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST
","id":503,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","stdout":"","subt
asks":[],"time":0,"type":0}, {"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host
":"NODE_HOST","id":504,"node":"NODE_NAME","pid":0,"port":"NODE_PORT","state":8,"stderr":"","st
dout":"","subtasks":[],"time":0,"type":0}], {"error_code":0,"error_message":"","exit_s
tatus":0,"fields":[],"files":[],"host":"NODE_HOST","id":502,"node":"NODE_NAME","pid":0,"port":"NODE
_PORT","state":2,"stderr":"","stdout":"","subtasks":[{"error_code":0,"error_message":"","exit_status"
:0,"fields":[],"files":[],"host":"NODE_HOST","id":505,"node":"NODE_NAME","pid":0,"port":"NODE_PORT
","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0}, {"error_code":0,"error_message":"","
","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":506,"node":"NODE_NAME","pid":0,"port":
"NODE_PORT","state":8,"stderr":"","stdout":"","subtasks":[],"time":0,"type":0}], {"time":0,"type":0}], "ti
me":103,"type":0}]
```

Example 13

//Request protocol

//Request example set sync task to execute request use sync subtasks

```
{"data":"eyJjb21tYW5kljoic29ydCIsImZpbGVzIjpwXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsa
W1pdHMiOnsiY3B1IjowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\nb2NfbWF4IjowLCJyc
mFtX2ZyZWUiOjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\nljowLCJ2cmFtX2ZyZWUiOjAsInZy
YW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uIjpw7ImNsZWFu\r\nndXAiOjAsImVudmlyb25tZW50IjpwXSwiaG9tZV9
kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwi
dGltZW91dCI6MCwi\r\nndG1vZGUiOjEsInR5cGUiOjAsInVzZXliOiliX0=","id":500,"subtasks":[{"data":"eyJjb
21tYW5kljoic29ydCIsImZpbGVzIjpwXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsi
Y3B1IjowLCJkaXNrX2ZyZWUiOjAsImRpc2tfZnJlZV9taW4iOjAsInBy\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUi
OjAsInJyYW1fZnJlZV9taW4iOjAsInRocmVhZHNfbWF4\r\nljowLCJ2cmFtX2ZyZWUiOjAsInZyYW1fZnJlZV9t
aW4iOjB9LCJzZXNzaW9uIjpw7ImNsZWFu\r\nndXAiOjAsImVudmlyb25tZW50IjpwXSwiaG9tZV9kaXJlY3Rvcnki
OiliLCJwYXNzd29yZCI6\r\nliIsInBvcnQiOjAsInNoZWxsljoiliwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6
MCwi\r\nndG1vZGUiOjEsInR5cGUiOjAsInVzZXliOiliX0=","id":501,"subtasks":[{"data":"eyJjb21tYW5kljoic2
9ydCIsImZpbGVzIjpwXSwiaW5wdXQiOiJkZWZcbmFiY1xuMTIzXG4i\r\nLCJsaW1pdHMiOnsiY3B1IjowLCJka
```

XNrX2ZyZWUiOjAslmRpc2tfZnJlZV9taW4iOjAslnBy\r\nb2NfbWF4IjowLCJycmFtX2ZyZWUiOjAslnJyYW1fZnJlZV9taW4iOjAslnRocmVhZHNfbWF4\r\nlJowLCJ2cmFtX2ZyZWUiOjAslnZyYW1fZnJlZV9taW4iOjB9LCJzZXNzaW9uljpb7ImNsZWFu\r\nndXAIjOjAslmVudmlyb25tZW50IjpbXSwiaG9tZV9kaXJlY3RvcnkiOiliLCJwYXNzd29yZCI6\r\nnliIslnBvcnQlOjAslnNoZWxsljoiIiwidGltZV9tYXgiOjIwMDAwMCwidGltZW91dCI6MCwi\r\nndG1vZGUlOjEslnR5cGUlOjAslnVzZXiOiIiifX0=","id":503,"subtasks":[],"type":0},"data": "eyJjb21tYW5kIjoic29yZCI6MCwiLCkKaXN... (repeated many times with varying IDs)

/Response protocol

//Response example set sync task to execute request use sync subtasks

```
{{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":500,"node":"NODE_NAME","pid":5283,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTlzMmFiYwpkZwYk","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":501,"node":"NODE_NAME","pid":5285,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTlzMmFiYwpkZwYk","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":503,"node":"NODE_NAME","pid":5287,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTlzMmFiYwpkZwYk","subtasks":[],"time":104,"type":0},"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":504,"node":"NODE_NAME","pid":5289,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTlzMmFiYwpkZwYk","subtasks":[],"time":104,"type":0}],{"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":502,"node":"NODE_NAME","pid":5291,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTlzMmFiYwpkZwYk","subtasks":[{"error_code":0,"error_message":"","exit_status":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":505,"node":"NODE_NAME","pid":5285,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTlzMmFiYwpkZwYk","subtasks":[],"time":104,"type":0},"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":506,"node":"NODE_NAME","pid":5291,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTlzMmFiYwpkZwYk","subtasks":[],"time":104,"type":0}]}
```



```
s":0,"fields":[],"files":[],"host":"NODE_HOST","id":505,"node":"NODE_NAME","pid":5293,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTIzCmFiYwpkZWYK","subtasks":[],"time":104,"type":0},{  
"error_code":0,"error_message":"","exit_status":0,"fields":[],"files":[],"host":"NODE_HOST","id":506,"node":  
"NODE_NAME","pid":5295,"port":"NODE_PORT","state":0,"stderr":"","stdout":"MTIzCmFiYwpkZWYK"  
,"subtasks":[],"time":104,"type":0}],"time":104,"type":0}],"time":104,"type":0}}
```