Terminal Management API

Version 2.0.6

Content

General description

Control interface consists of two parts: Requests WebSocket at **:8765/request** and Updates WebSocket at **:8765/updates.** Exchange is facilitated via the text passed through the WebSockets. All command responses and notifications are sent only in JSON format. Terminal updates are encrypted and encoded using Base64. Encryption key can be received in the process of authorization.

Command format

All commands are sent in JSON format, while parameters, if any, should be encoded using Base64 (in order to correct the scripts including Latin symbols, special characters, etc.).

Example:

```
{
  "method" : "command",
  "uid" : "z3NIxqvK0C328hRi",
  "script" : "getAppState()"
}
{
  "method" : "auth",
  "version" : "1.0",
  "mechanism" : "PLAIN",
  "login" : "admin",
  "password" : "admin"
}
```

Parameters description

- method – the name of the method. Each method has its own parameter list. Each command has a returned value in JSON format. There are the following methods:
1) command – a command execution request. It includes the following parameters:
    - script – script execution start.

- scope – defines if the script should be executed in a global scope or local scope.

- uid – a personal key which is used to check if the incoming command is an authorized user request and is generated during authorization.

Possible interaction errors:

- incorrect command type:

```
{
  "error" : "unknown method",
  "event" : "request"
}
```

- incorrect request format:

```
{
  "error" : "api error: method-command?script=getAppState()&uid=123",
  "event" : "request"
}
```

- internal handler error. Lua will return you the message indicating an error type. For example, if you set acceptPeer command instead of acceptPeer, the following error will appear:

```
{
  "error" : "attempt to call global 'acceptPee' (a nil value)",
  "event" : "commandExecution"
}
```

- method parameter is absent; in case it is present, method parameter is not string type.

```
{
  "error" : "'method' param must be present and must be string type",
  "event" : "request"
}
```

- one of the necessary parameters is absent.

```
{
  "error" : "all params must be present",
  "event" : "request"
}
```

- the following error appears when one of the parameters is not string type.

```
{
  "error" : "all params must be string type",
  "event" : "request"
```

}
- wrong parameter of the scope where the script is to be executed.

```
{
  "error" : "wrong scope type",
  "event" : "request"
}
```

2) auth - authorization request. Format and procedure are described in authorization section.

3) sandbox - a parameter which indicates script execution scope. It allows users to create/delete their personal sandboxes. Your personal scope will be removed automatically when not in use for more than 30 min. Each time you execute the script in local network, the inaction time is reset. Examples:

```
{
  "method" : "sandbox",
  "uid" : "bccyqwDGuNR5dRwz",
  "task" : "create"
}
```

Parameters Description

- task – an indication of the action. create is to create your scope, delete is to remove your scope.
- uid – your personal "key" to check if an inbound command is a request from authorized user. Uid is received in the process of authorization.

Example of successful execution:

```
{
  "create" : "ok",
  "event" : "sandbox"
}
```

Possible execution errors:

- An attempt to work with the scope when the uid has expired or is unreadable:

```
{
    "error" : "your uid is invalid or out of date",
    "event" : "sandbox"
}
```

- An attempt to work with the scope by indicating task different from the permitted parameters (either create or delete):

```
{
     "error" : "wrong sandbox command",
     "event" : "sandbox",
     "sandbox" : "failure",
}
```

- An attempt to remove your personal scope if it has not been created before:

```
{
    "error" : "you don't have a sandbox,
    "event" : "sandbox",
    "delete" : "failure",
}
```

- An attempt to create your personal scope if it has been created before:

```
{
   "create" : "failure",
   "error" : "you already have your own sandbox",
   "event" : "sandbox"
}
```

- A necessary parameter is absent:

```
{
   "create" : "failure",
   "error" : "all params must be present",
   "event" : "sandbox"
}
```

- An error occurring when one of the parameters is not string type:

```
{
   "create" : "failure",
   "error" : "all params must be string type",
  "event" : "sandbox"
}
```

Authorization process

Before you start working with the client, you need to authorize. Currently the client supports ZMTP Plain Security Mechanism. The ZMTP Plain Security Mechanism defines a simple username/password mechanism that lets the server authenticate a client on the server and is used in internal networks where security requirements are low. All personal details including username, password, and uid are unencrypted.

You can have the following authorization privileges:

- Administrator who has a full control over the terminal;
- Standard user who can make calls, create conferences, and change

hardware setup.

Authorization mode is set automatically in the process of authorization depending on your username and password. If you are already logged in and try to authorize again with the same set of privileges, you will be notified that you already have the key and have been authorized. If you have successfully authorized with a different set of privileges, you will be provided with a new key and a new set of privileges. During you first authorization as an administrator, you will have to reset admin password. Until then you will be blocked from command execution. The following error will be identified:

```
{
  "error" : "you must reset admin password first",
  "event" : "commandExecution",
  "getAppState" : "failure"
}
```

Authorization mechanism:

1) the client receives a request such as:

```
{

  "method" : "auth",
  "version" : "1.0",
  "mechanism" : "PLAIN",
  "login" : "admin",
  "password" : "admin"

}
```

Where all the specified fields are mandatory and indicate the following:

- auth – a method name which indicates authorization request.
- version – an authorization protocol version (currently version 1.0).
- login – a plain login for authorization.
- mechanism – authorization mechanism (currently PLAIN).
- password – an authorization password.

2) If the authorization has been successful, the terminal will send you uid which will help you interact with the terminal and a key which will help you decrypt notifications. Uid becomes obsolete after 30 min of inaction. Each time you access the server, uid inaction time is reset again for 30 minutes. If your uid has been removed, your sandbox will also be removed.

A sample of response you receive if the authorization has been successful:

```
{
  "auth" : "ok",
  "event" : "auth",
  "key" :
"90f3b7f99d34eb401f774d16284b92fcd10e815c9fd710f6fc61b599a980e129
",
  "previleges" : 1,
  "warning" : "you should set new administrator password. The following
commands will be blocked until this is undone"
  "uid" : "fjpQFlAH03WOeqI3",
  "cid" : "sadasdasdddasfsd"
}
```

Where

- Auth - a command result.
- Event - a field indicating the type of message.
- Key - used to decrypt notifications.
- Uid - a unique user id.
- cid - a unique client user ID which is used in notifications.
- warning - a field which is shown only if you are logged in as an administrator. The field notifies you that you should reset admin password. Until then all the commands will be blocked.
- Privileges is an authorization mode. It could be
- * 0 - standard user
- * 1 - administrator

3) This step is necessary only if you are logged in as an administrator. At this point you should reset admin password. Until then all your commands will be blocked.

```
{
  "error" : "you must reset admin password first",
  "needResetPassword" : "true",
  "event" : "commandExecution",
  "getAppState" : "failure"
}
```

You can reset your password via setAuthParams command.

Possible authorization errors:

- Wrong protocol version:

```
{
  "error" : "wrong protocol version - server uses 1.0",
  "auth" : "failure",
```

```
  "event" : "auth",
  "type" : 0
}
```

- Wrong security mechanism:

```
{
  "error" : "wrong security mechanism - server uses PLAIN",
  "auth" : "failure",
  "event" : "auth",
  "type" : 1
}
```

- Wrong username or password:

```
{
  "error" : "wrong username or password",
  "auth" : "failure",
  "event" : "auth",
  "type" : 2
}
```

- Request for command execution, slideshow, etc. with invalid or out of date uid.:

```
{
  "error" : "your uid is invalid or out of date",
  "event" : "commandExecution",
  "type" : 5,
}
```

- Re-authorization of the user who has already logged in:

```
{
  "auth" : "failure",
  "error" : "you already have uid",
  "event" : "auth",
  "key" :
"3aef695a838f222cbf30fcd3046224e7bd00b022dca9a306f5d03f4a255b62cf"
,
  "cid" : "asdaffdsfsdfsdfsdf",
  "type" : 3,
  "previleges" : 0,
  "uid" : "fjpQFlAH03WOeqI3",
  "cid" : "asdasdasd"
}
```

Parameter Description

- Key – AES-256 key for notification decryption displayed in hexadecimal form.

- Uid – your unique user id used for command execution.
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- Auth – a result of command execution. If the command execution is successful, it is 1. Otherwise, it is 0. If it is 0, error field with error description will be displayed.
- Error – a field indicating error text.
- needResetPassword - a special field indicating a need to reset admin password for further work.
- cid – a unique client parameter within the application which is used for notifications.
- Type – an error type
- Privileges – an authorization mode. It could be
  * 0 - standard user
  * 1 - administrator

Management commands

accept

Format: accept()
Example: accept()
Description: Accept the call. The command is run immediately and the result of execution is received at once.
Parameter description:
- event – event or command indicator. It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- accept – command name or the result of its execution. If it is successful, it is "ok", otherwise it is "failure". In case of failure you will be shown an error field with error description.

Response example:

```
{
  "event" : "commandExecution",
  "accept" : "ok"
}
```

Possible execution errors:

- An attempt to receive conference call when nobody is calling you:

```
{
  "error" : "nobody calling you",
  "accept" : "failure",
  "event" : "commandExecution"
}
```

- Unexpected error. Try again:

```
{
```

```
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "accept" : "failure"
}
```

- An attempt to receive the request to participate in active conference by the “accept” command. In this case you need to use accepPeer() command:

```
{
  "error" : "you should use 'acceptPeer()' in conference",
  "event" : "commandExecution",
  "accept" : "failure"
}
```

- An attempt to use “accept” command if you make a call:

```
{
  "error" : "you should use 'hangUp()' in this case",
  "event" : "commandExecution",
  "accept" : "failure"
}
```

- An attempt to use “accept” command with parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "accept" : "failure"
}
```

- An attempt to use command which is not allowed for you:

```
{
  "error" : "accept is not allowed for you",
  "accept" : "failure",
  "event" : "commandExecution"
}
```

acceptPeer

Format: acceptPeer(base64PeerId)
Example: acceptPeer(base64_encode("ivanov@trueconf.com"))
Description: to accept a request from the user to participate in your conference. The command is run immediately and the result of execution is received at once.
Parameter description:
-event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.

- acceptPeer – an execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "acceptPeer" : "ok"
}
```

Possible execution errors:

- An attempt to accept a participant's request to join the conference if the terminal is not taking part in the conference:

```
{
  "error" : "you're not in conference",
  "acceptPeer" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to accept a participant's request to join the conference if you are not the conference owner:

```
{
  "error" : "you're not conference owner",
  "acceptPeer" : "failure",
  "event" : "commandExecution"
}
```

An attempt to accept a participant's request to join p2p call:

```
{
  "error" : "you can't do this in p2p conference",
  "acceptPeer" : "failure",
  "event" : "commandExecution"
}
```

- Unexpected error. Try again later:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "acceptPeer" : "failure"
}
```

- An attempt to run acceptPeer() command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "event" : "commandExecution",
  "acceptPeer" : "failure"
}
```

An attempt to run a command without permission:

```
{
  "error" : "acceptPeer is not allowed for you",
  "acceptPeer" : "failure",
```

```
  "event" : "commandExecution"
}
```

- An attempt to accept a participant's request to join the conference by indicating callid of the participant who is not calling you:

```
{
  "error" : "peerId is not found in request invite list",
  "acceptPeer" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command by indicating incomplete callId in the parameters:

```
{
   "acceptPeer": "failure",
   "error": "peerId must be in a full form",
   "event": "commandExecution"
}
```

acceptInvitationToPodium

Format acceptInvitationToPodium()
Example: acceptInvitationToPodium()
Description: Accept the invitation to take the podium. It is used only in a role-based conference after an invitation to take the podium has been sent by the conference moderator. At any time during the call the command will bring ok back and the roles will not be changed. The command is run immediately and the result of execution is received at once.

Parameters description:
-event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- acceptInvitationToPodium - an execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "acceptInvitationToPodium" : "ok",
  "event" : "commandExecution"
}
```

Possible execution errors:

- An attempt to accept the invitation if the terminal is not in the conference:

```
{
  "error" : "you're not in conference",
  "acceptInvitationToPodium" : "failure",
  "event" : "commandExecution"
```

```
}
```

- An attempt to accept the invitation in case the conference is not a role-based one:

```
{
  "error" : "you're not in role conference",
  "acceptInvitationToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to accept the invitation if you are the conference owner. In this case you should run gotoPodium() и leavePodium() commands, because you have permission to do it.

```
{
  "error" : "you're conference owner, use 'gotoPodium()', 'leavePodium()'
etc...",
  "acceptInvitationToPodium" : "failure",
  "event" : "commandExecution"
}
```

- Unexpected error. Try again later:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "acceptInvitationToPodium" : "failure"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "acceptInvitationToPodium" : "failure",
  "error" : "parameters must be empty",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "acceptInvitationToPodium is not allowed for you",
  "acceptInvitationToPodium" : "failure",
  "event" : "commandExecution"
}
```

acceptRequestToPodium

Format: acceptRequestToPodium()
Example: acceptRequestToPodium()
Description: allow the user who sent the request to take the podium. The command is run immediately and the result of execution is received at once.

Parameters description:
-event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- acceptRequestToPodium - an execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "acceptRequestToPodium" : "ok",
  "event" : "commandExecution"
}
```

Possible execution errors:

- An attempt to run the command if the terminal is not in the conference:

```
{
  "error" : "you're not in conference",
  "acceptRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command if the conference is not a role-based one:

```
{
  "error" : "you're not in role conference",
  "acceptRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command if you are not the conference owner:

```
{
  "error" : "you're  not conference owner",
  "acceptRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with parameters pre-filled:

```
{
  "acceptRequestToPodium" : "failure",
  "error" : "parameters must be empty",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "acceptInvitationToPodium is not allowed for you",
```

```
  "acceptRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when there are no corresponding requests:

```
{
  "error" : "there is no any requests",
  "acceptRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

activateLicense

Format: activateLicense(licenseKey)
Example: activateLicense("G3K3-E929-837P-BHNQ-GKAV-GSLH-T5YU-3TJ8-ECWD-YBRV-J7A2")
Description: Activate license key. Activation result will be returned to you immediately in response parameters.

Parameters description:
-event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- activateLicense - an execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- status – a license key status. It can be valid (the activation has been successful) or invalid (activation error).

Response example:

```
{
  "event" : "commandExecution",
  "activateLicense" : "ok",
  "status" : "valid"
}
```

Possible execution errors:

- An attempt to run activateLicense() command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "event" : "commandExecution",
  "activateLicense" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "activateLicense is not allowed for you",
  "activateLicense" : "failure",
```

```
  "event" : "commandExecution"
}
```

addContactToAbook

Format: addContactToAbook(base64EncodedCallId, base64EncodedDisplayName)
Description: Add a contact into the Address Book. The result will appear immediately.

Parameters description:
-event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- addContactToAbook - an execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- base64EncodedCallId – callId of the contact you need to add in BASE64.
- base64EncodedDisplayName - display name of the contact in BASE64.

Response example:

```
{
  "event" : "commandExecution",
  "addContactToAbook" : "ok"
}
```

Possible execution errors:

- An attempt to run the command addContactToAbook() with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "event" : "commandExecution",
  "addContactToAbook" : "failure"
}
```

- An attempt to run the command with incorrect parameters.

```
{
  "error" : "check parameters",
  "addContactToAbook" : "failure",
  "event" : "commandExecution"
}
```

allowRecord

Format: allowRecord()
Example: allowRecord()
Description: Allow recording of your video stream. The command is run

immediately and the result of execution is received at once.

Parameter description:
-event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- allowRecord - an execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

Response example:

```
{
  "allowRecord" : "ok",
  "event" : "commandExecution"
}
```

Possible execution errors:

- An attempt to run the command if the terminal is not in the conference:

```
{
  "error" : "you're not in conference",
  "allowRecord" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when there are no corresponding requests:

```
{
  "error" : "there is no any requests",
  "allowRecord" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with pre-filled parameters:

```
{
  "allowRecord" : "failure",
  "error" : "parameters must be empty",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "allowRecord is not allowed for you",
  "allowRecord" : "failure",
  "event" : "commandExecution"
}
```

appUpdate

Format: appUpdate()
Example: appUpdate()
Description: To start client application update. If the client has been successfully updated, the client application will automatically be reset.
Parameters Description:
-event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- appUpdate - an execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

Response example:

```
{
  "appUpdate" : "ok",
  "event" : "commandExecution"
}
```

Possible execution errors:

- An attempt to run the command with pre-filled parameters:

```
{
  "appUpdate" : "failure",
  "error" : "parameters must be empty",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "appUpdate is not allowed for you",
  "appUpdate" : "failure",
  "event" : "commandExecution"
}
```

Call

Format: call("base64PeerId@some.server")
Example: call("base64_encode(ivanov@trueconf.com")
Description: to call the user with peerId. This command makes p2p call. If callId exceeds the maximum length, it will be shortened and the warning field will be displayed. Ok means the command has been accepted for the execution but it has not been successfully run yet. Execution result will come out via notification.

Parameter description
- peerId - a unique user ID (TrueConfID)
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.

- call - a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "call" : "ok",
  "warning" : "callee user was too long, so it is cutted to default maximum length"
}
```

Possible execution results:

- An attempt to call the user if the application is not in the state 3 (see getAppState() command), which means it is in the conference, is calling, etc.

```
{
  "error" : "can't do call, check application state",
  "event" : "commandExecution",
  "call" : "failure"
}
```

- An attempt to call by indicating your unique user ID (TrueConfID) in the parameters:

```
{
  "error" : "you can't call to yourself",
  "call" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to call by indicating empty user ID (TrueConfID) in the parameters:

```
{
  "error" : "peerId is empty",
  "call" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "call" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to indicate an incomplete peerId

```
{
  "call" : "failure",
  "error" : "peerId must be in a full form",
```

```
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "call is not allowed for you",
  "call" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when your account does not have permission:

```
{
  "error" : "not enough rights",
  "call" : "failure",
  "event" : "commandExecution"
}
```

changeVideoMatrix

Format: changeVideoMatrix("base64_encode(JsonMatrix")
Description: specify video matrix and the ratio of video windows for available slots.
It is used only in the conference. Receiving 'ok' means the command has been accepted for execution but has not been executed yet. You will receive execution results via **onChangeVideoMatrixReport** notification. To hide/show a user in the layout, you need to list or remove a user into transmitted participants list. A user in slotId: 1 will be active. slotId number can range from 1 to 27 and must start with 1.

Sample request: changeVideoMatrix("base64_encode({"matrixType" : 1})") indicates allocation matrix (one big video and smaller ones)
JSON sample with participants list included

```
{
  "matrixType" : 1,
  "participants" : [
    {
      "slotId" : 1,
      "peerId" : "ivanov@trueconf.com"
    },
    {
      "slotId" : 2,
      "peerId" : "Ivan_IV@trueconf.com"
    },
    {
      "slotId" : 3,
      "participant" : "IV@trueconf.com"
    }
```

```
  ]
}
```

Parameters description

- event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- changeVideoMatrix – a result of sending the command to be executed. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.
- participants – the list of video slots and conference participants
- slotId – slot number
- participant – a conference participant ID
- matrixType – a matrix allocation type. There are the following matrix types:
    * even= 0, all the windows are of the same size (for multipoint conference)
    * big = 1, one window is big while others are small (for multipoint conference)
    * one = 2, display only the video of the conference participant who is the first in the participants list (for any type of the conference)
    * oneSelf= 3, big video of the conference participant and a small self-view in the corner (for video call)

Response example:

```
 {
  "event" : "commandExecution",
  "changeVideoMatrix" : "ok"
 }
```

Possible execution error:

- Invalid matrixType parameter type:

```
{
  "error" : "matrixType must be integral type",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- Required peerId parameter is absent:

```
{
  "error" : "peerId must be present",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- Required slotId parameter "" is absent:

```
{
  "error" : "slotId must be present",
```

```
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- "peerId" parameter is not string type:

```
{
  "error" : "peerId must be string type",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- "slotId" parameter type is not numeric:

```
{
  "error" : "slotId must be integral type",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- Required parameter "matrixType"is absent:

```
{
  "error" : "matrixType must be present",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- Incorrect JSON format:

```
{
  "error" : "Bad JSON format",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- specified participant list is not array:

```
{
  "error" : "participant list is not array",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you are not in the conference:

```
{
  "error" : "you're not in conference",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- Wrong matrixType parameter value:

```
{
  "error" : "bad matrix type",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- You indicated several identical peerID in the participant list:

```
{
  "error" : "same peers",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- You indicated empty peerId:

```
{
  "error" : "peerId must be not empty",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- You indicated peerId of the user who is currently not in the conference:

```
{
  "error" : "peerId 'ivanov@trueconf.com' not found",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- You indicated several identical slot numbers:

```
{
  "error" : "same slot numbers",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to indicate the number of slots which exceeds the maximally permitted:

```
{
  "error" : "too many slots for displaying",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
}
```

- Incorrect slot numbers order:

```
{
  "error" : "bad slot numbers order",
  "changeVideomatrix" : "failure",
  "event" : "commandExecution"
```

```
}
```

changeWindowState

Format: changeWindowState(windowType, stayOnTop),
Example: changeWindowState(2, true),
Description: set video window features. The result of execution is received at once.

Parameter description:
- event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- changeWindowState – command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- windowType – window feature, it can be
    * 1 = minimized;
    * 2 = full screen mode.
- stayOnTop – Required field. It shows of the window overlays other windows. It can be
    * true
    * false

Response example:

```
{
  "event" : "commandExecution",
  "changeWindowState" : "ok"
}
```

Possible execution errors:

- An attempt to run the command without permission:

```
{
  "error" : "changeWindowState is not allowed for you",
  "changeWindowState : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "changeWindowState : "failure",
  "event" : "commandExecution"
}
```

- An attempt to specify windowType parameter not as int:

```
{
  "error" : "windowType must be int",
  "changeWindowState : "failure",
  "event" : "commandExecution"
}
```

- An attempt to specify stayOnTop parameter not as bool:

```
{
  "error" : "stayOnTop must be bool",
  "changeWindowState : "failure",
  "event" : "commandExecution"
}
```

connectToServer

Format: connectToServer("base64_encode(server")**,** where server line has the following format: serv_hostname[:serv_port] | **@serv_domain**
Example: connectToServer("base64_encode("111.111.111.111:1234")), connectToServer()
Description: connect to the server or TrueConf Online service via defined IP/ hostname:port. Receiving 'ok' means the command has been accepted for execution but has not been executed successfully yet. The execution result will be received via notification.

- If the port is not defined, the port will be used by default.
- If it is empty, it is connected to the same place it was connected previously.
- If it starts with @, it will be connected to the DNS service.
- serv_domain

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- connectToServer – the result of sending the command for execution
It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

```
{
  "event" : "commandExecution",
  "connectToServer" : "ok"
}
```

Possible execution errors:

- An attempt to use the command without permission

```
{
  "error" : "connectToServer is not allowed for you",
  "connectToServer" : "failure",
  "event" : "commandExecution"
```

```
}
```

createConference

Format: createConference("base64_encodedName", "base64_encodedType", "base64_encodedAutoAccept", "base64_encodedUsers")
Example:
- createConference("base64_encode(Monthly report)", "base64_encode(symmetric)", "base64_encode(false)", "base64_encode(ivanov@trueconf.com, petrov@trueconf.com)"),
Description: to create the conference with specified parameters and participants. name and users parameters can be empty. If the latter parameter is absent, the conference with only one participant (you) will be created. If the conference title is too long, it will be cropped. In response you will receive warning field. You can get current maximum length by running **getMaxConfTitleLength**. However, you still can invite other users into the conference. Receiving ok means the command has been accepted for execution but has not been successfully executed yet. The execution result will be received via notification.

Parameter description:
- base64_encodeName – conference title
- base64_encodeAutoAccept – an indicator which gives permission to automatically accept participants into the conference.
- base64_encodeType – Conference type. It can be
  * symmetric - symmetric
  * asymmetric – assymetric
  * role – role-based
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- createConference - command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

Response example:
```
{
  "event" : "commandExecution",
  "createConference" : "ok",
  "warning" : "title was too long, so it is cutted to default maximum length"
}
```

Possible execution result:
- You are trying to create incorrect type of the conference:

```
{
  "error" : "Wrong conference type",
  "createConference" : "failure",
  "event" : "commandExecution"
```

```
}
```

- An attempt to create a conference when you are already in the conference:

```
{
  "error" : "can't create conference, check application state",
  "createConference" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to create a conference unavailable for your plan:

```
{
  "error" : "Your current tariff doesn't allow you to create this type of
conference",
  "createConference" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "createConference" : "failure",
  "event" : "commandExecution"
}
```

- You defined ‘type’ parameter not as a string parameter:

```
{
  "error" : "conference type must be string",
  "createConference" : "failure",
  "event" : "commandExecution"
}
```

You indicated incomplete { "createConference" : "failure", "error" : "peerId must be in a full form", "event" : "commandExecution" }

- An attempt to run the command when you do not have the rights:

```
{
  "error" : "createConference is not allowed for you",
  "createConference" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the conference with the number of participants exceeding the allowed maximum participants number for this conference type:

```
{
  "error" : "participants count is bigger then allowed maximum participants
number for this conference type",
  "createConference" : "failure",
```

```
  "event" : "commandExecution"
}
```

- An attempt to create a conference without permission:

```
{
  "error" : "not enough rights",
  "createConference" : "failure",
  "event" : "commandExecution"
}
```

denyRecord

Format: denyRecord()
Example: denyRecord()
Description: prohibit your video stream recording. The command is run immediately and the result of execution is received at once.

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- denyRecord – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

Response example:

```
{
  "denyRecord" : "ok",
  "event" : "commandExecution"
}
```

Possible execution errors:

- An attempt to run the command when the terminal is not in the conference:

```
{
  "error" : "you're not in conference",
  "denyRecord" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when there are no corresponding requests:

```
{
  "error" : "there is no any requests",
  "denyRecord" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with pre-filled parameters:


```
{
  "denyRecord" : "failure",
  "error" : "parameters must be empty",
  "event" : "commandExecution"
}
```


- An attempt to run the command without permission:


```
{
  "error" : "denyRecord is not allowed for you",
  "denyRecord" : "failure",
  "event" : "commandExecution"
}
```


extendUidTtl

Format: extendUidTtl()
Example: extendUidTtl()
Description: renew your uid

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- extendUidTtl – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

Response example:


```
{
  "extendUidTtl" : "ok",
  "event" : "commandExecution"
}
```


Possible execution errors:

- An attempt to run the command with pre-filled parameters:


```
{
  "extendUidTtl" : "failure",
  "error" : "parameters must be empty",
  "event" : "commandExecution"
}
```


- An attempt to run the command without permission:


```
{
  "error" : "extendUidTtl is not allowed for you",
  "extendUidTtl" : "failure",
  "event" : "commandExecution"
}
```

getAbook

Format: getAbook()
Example: getAbook()
Description: receive Address Book. The command is run immediately and the result of execution is received at once.

Parameter description:
- peerId – unique user ID (*TrueConfID*)
- peerDn – display user name which is encoded in BASE64 sequence. It needs reverse decoding.
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- status – user's status which can have the following values:
  * USER_INVALID = -1,
  * USER_LOGOFF = 0,
  * USER_AVAIL = 1,
  * USER_BUSY = 2,
  * USER_MULTIHOST = 5
- getAbook – command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

Response example:

```
{
  "abook" : [
    {
      "peerId" : "demo_conf@trueconf.com",
      "peerDn" : "Group conference demo",
      "status" : 5
    },
    {
      "peerId" : "echotest_ru@trueconf.com",
      "peerDn" : "Echo test",
      "status" : 0
    },
  "event" : "commandExecution",
  "getAbook" : "ok"
  ]
}
```

Possible execution errors:

- An attempt to view contact list when you are not logged in:

```
{
  "error" : "you should login first",
  "getAbook" : "failure",
  "event" : "commandExecution"
```

```
}
```
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getAbook" : "failure"
}
```
- An attempt to run the command without permission:

```
{
  "error" : "getAbook is not allowed for you",
  "getAbook" : "failure",
  "event" : "commandExecution"
}
```

getAppState

Format: getAppState()
Example: getAppState()
Description: return application state. The command is run immediately and the result of execution is received at once.

Response example:

```
{
  "appState" : 3,
  "event" : "commandExecution",
  "getAppState" : "ok",
  "key" :
"90f3b7f99d34eb401f774d16284b92fcd10e815c9fd710f6fc61b599a980e129
",
}

{
  "appState" : 4,
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com",
  "event" : "commandExecution",
  "getAppState" : "ok",
  "key" :
"90f3b7f99d34eb401f774d16284b92fcd10e815c9fd710f6fc61b599a980e129
",
  "waitDir" : "incoming",
  "waitType" : "p2p",
}
```

Parameter description:
- appState – application state which can have the following values:
  * none = 0 (No connection to the server and the terminal does nothing),
  * connect = 1 (the terminal tries to connect to the server),

  * login   = 2 (you need to login),
  * normal   = 3 (the terminal is connected to the server and logged in),
  * wait   = 4 (the terminal is pending: either it calls somebody or somebody calls it),
  * conference = 5 (the terminal is in the conference),
  * close   = 6 (the terminal is finishing the conference).

- peerId – parameter is present only during wait state and defines TrueConfID, a unique ID of the user who calls you, or the user you are calling.
- peerDn – parameter is present only during wait state and defines display username who calls you or the user you are calling. It is encoded in Base 64 sequence and needs reverse decoding.
- key – a key for notifications decoding.
- slideShowPresent – slideshow indicator. It can be true or false.
- displayName – display name of the authorized user (you).
- conferenceOwner – the parameter is present only of the terminal is in the group conference and indicates the conference owner.
- confType – the parameter is present only if the terminal is currently in the session and defines its type. It can be:
  * p2p   = 0
  * symmetric = 1
  * asymmetric = 2
  * role   = 3
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getAppState – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- waitDir – defines the direction of the call. It can be
  * outgoing (indicates that you are calling someone)
  * incoming (indicates that you are being called)
- waitType – display call type. It can be
  * p2p – video call;
  * conference – group conference;
- embeddedHttpPort - port number matching embedded http server which gives away conference frames. If the server is not started it is 0.

Possible execution errors:

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getAppState" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getAppState is not allowed for you",
```

```
  "getAppState" : "failure",
  "event" : "commandExecution"
}
```

getAudioMute

Format: getAudioMute()
Example: getAudioMute()
Description: to get to know if the audio is switched off. The command is run immediately and the result of execution is received at once.

Response example:

```
{
  "event" : "commandExecution",
  "getAudioMute" : "ok",
  "mute" : true
}
```

Parameter description
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getAudioMute – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- mute – indicates if the audio is switched off.

Possible execution errors:

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getAudioMute" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getAudioMute is not allowed for you",
  "getAudioMute" : "failure",
  "event" : "commandExecution"
}
```

getBroadcastSelfie

Format: getBroadcastSelfie()
Example: getBroadcastSelfie()
Description: get to know if you can view the video using current camera. The command is run immediately and the result of execution is received at

once.

Response example

```
{
  "event" : "commandExecution",
  "getBroadcastSelfie" : "ok",
  "enabled" : true,
  "fps" : 32
}
```

Parameter description:
- enabled – parameter which indicates if current functions are included.
- fps – FPS number.
- event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getBroadcastSelfie - is a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Possible execution errors:

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getBroadcastSelfie" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getBroadcastSelfie is not allowed for you",
  "getBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

getDisplayNameById

Format: getDisplayNameById(base64PeerId)
Example: getDisplayNameById(base64_encode("ivanov@trueconf.com"))
Description: get display name using user ID. The command is run immediately and the result of execution is received at once:

Response example:

```
{
  "event" : "commandExecution",
  "getDisplayNameById" : "ok",
  "displayName" : "Ivan Ivanov"
}
```

Parameter description:
- displayName – display name of the requested user. If display name of the requested user failed to be received, the command with return callId .
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getDisplayNameById – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Possible execution errors::
- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must not be empty",
  "event" : "commandExecution",
  "getDisplayNameById" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getDisplayNameById is not allowed for you",
  "getDisplayNameById" : "failure",
  "event" : "commandExecution"
}
```

getConferenceParticipants

Format: getConferenceParticipants()
Example: getConferenceParticipants()
Description: to view conference participants list. The command is run immediately and the result of execution is received at once.

Response example:
For video call:

```
{
  "Participants" : [
    {
      "peerDn" : "Ivan Ivanov",
      "peerId" : "ivanov@trueconf.com",
      "mic" : true,
      "video" : false
    },
    {
      "peerDn" : "Petr Petrov",
      "peerId" : "petrov@trueconf.com",
      "mic" : false,
      "video" : false
    }
  ],
```

```
  "event" : "commandExecution",
  "getConferenceParticipants" : "ok"
}
```
For group conference:
```
{
  "Participants" : [
    {
      "broadcast" : true,
      "peerDn" : "Ivan Ivanov",
      "peerId" : "ivanov@trueconf.com",
      "role" : 1,
      "slideShowFlag" : false,
      "mic" : false,
      "video" : false
    },
    {
      "broadcast" : true,
      "peerDn" : "Petr Petrov",
      "peerId" : "petrov@trueconf.com",
      "role" : 2,
      "slideShowFlag" : false,
      "mic" : true,
      "video" : true
    }
  ],
  "event" : "commandExecution",
  "getConferenceParticipants" : "ok"
}
```

Parameter description:

- peerId – a unique user ID (*TrueConfID*).
- peerDn – a display username which is encoded in BASE64 sequence. It needs reverse decoding (*TrueConfID*).
- broadcast – indicates if the user is taking the podium at the moment.
- mic – indicates microphone status. It can be true (turned on) или false (turned off)
- video – indicates camera status. It can be true (turned on) или false (turned off).
- slideShowFlag – indicates if the conference participant has started slideshow. It can be true or false.
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getConferenceParticipants – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- role – participant's role. It can be the following:
  * PR_COMMON = 1 (general),

* PR_LEADER = 2 (conference owner),
* PR_REPORTER = 4 (the user who is taking the podium),
* PR_REMARK = 5 (the user who is making an audio remark)

Possible execution errors:

- An attempt to view conference participants if the terminal is out of the conference:

```
{
  "error" : "You're not in conference",
  "event" : "commandExecution",
  "getConferenceParticipants" : "failure"
}
```

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getAppState" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getAppState is not allowed for you",
  "getAppState" : "failure",
  "event" : "commandExecution"
}
```

getContactDetails

Format: getContactDetails(base64_encode(peerId))
Example: getContactDetails(base64_encode("ivanov@trueconf.com"))
Description: get contact's personal details

Response example:
For video call:

```
{
   "getContactDetails": "ok",
   "event": "commandExecution"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getContactDetails – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Possible execution errors:

- An attempt to view conference participants if the terminal is not in the conference.

- An attempt to run the command with no parameters specified:


```
{
   "error": "parameters must be not empty",
   "getContactDetails": "failure",
   "event": "commandExecution"
}
```


- An attempt to run the command without permission:


```
{
  "error" : "getContactDetails is not allowed for you",
  "getContactDetails" : "failure",
  "event" : "commandExecution"
}
```


- An attempt to run the command with empty peerId:


```
{
   "getContactDetails": "failure",
   "error": "peerId must be not empty",
   "event": "commandExecution"
}
```


- An attempt to get personal details of the user who is not added into the Address Book:


```
{
   "getContactDetails": "failure",
   "error": "contact in not in your address book",
   "event": "commandExecution"
}
```


- peerId is not complete:


```
{
   "getContactDetails": "failure",
   "error": "peerId must be in a full form",
   "event": "commandExecution"
}
```


getLastSlide

Format: getLastSlide()
Example: getLastSlide()
Description: get the last slide. The response will contain full information about the slide: width, height, coding type, source code, etc.

Response example:


```
{
```

```
  "encode" : "base64",
  "event" : "commandExecution",
  "hash" :
"f2f8340eb41c3ea6c52c69a2aaeb55f5d2916fc23a9ab92d4f8f70d21f11cdb7",
  "height" : 50,
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com",
  "source" : "some base64 source of the image",
  "title" : "last-image.jpg",
  "width" : 50
}
```


Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getCurrentSlide – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.
- peerId – unique ID TrueConfID of the user who sent the slide
- peerDn – display user name which is encoded in BASE64 sequence. It needs reverse decoding
- source – BASE64 source
- hash – SHA256 BASE64 source hash to check if the image is transmitted right.
- encode – image coding type. It is currently supporting only BASE64.
- width - image width in pixels.
- height – image height in pixels.
- title – image title.

Possible execution errors:

- An attempt to run the command with pre-filled parameters:


```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getLastSlide" : "failure"
}
```


- An error which appears when none of the slides has been selected:


```
{
  "error" : "there is no last slide",
  "event" : "commandExecution",
  "getLastSlide" : "failure"
}
```


- An attempt to run the command without permission:


```
{
  "error" : "getLastSlide is not allowed for you",
  "getCurrentSlide" : "failure",
```

```
  "event" : "commandExecution"
}
```


getHardware

Format: getHardware()
Example: getHardware()
Description: get the list of hardware. The command is run immediately and the result of execution is received at once.

Response example:


```
{
  "audioCapture" : [
    {
      "description" : "alsa_input.usb-046d_0823_45871D00-00-
U0x46d0x823.analog-stereo",
      "name" : "0823 Analog Stereo"
    },
    {
      "description" : "alsa_input.pci-0000_00_1b.0.analog-stereo",
      "name" : "Built-in Audio Analog Stereo"
    }
  ],
  "audioCaptureName" : "default",
  "audioPlatform" : [
    {
      "name" : "ALSA"
    },
    {
      "name" : "PulseAudio"
    }
  ],
  "audioPlatformName" : "PulseAudio",
  "audioPlayback" : [
    {
      "description" : "alsa_output.pci-0000_00_1b.0.analog-stereo",
      "name" : "Built-in Audio Analog Stereo"
    }
  ],
  "audioPlaybackName" : "default",
  "event" : "commandExecution",
  "getHardware" : "ok",
  "videoCapture" : [
    {
      "description" : "/dev/video0",
      "name" : "UVC Camera (046d:0823)"
    }
  ],
  "videoCaptureName" : "default"
```

```
}
```

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getHardware – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- audioCapture – audio capture device.
- audioCaptureName – active audio capture device.
- audioPlayback – audio playback device.
- audioPlaybackName – active audio playback device.
- videoCapture – video devices.
- videoCaptureName – active video devices.
- name – device name.
- description – device description.

Possible execution errors:
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getHardware" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getHardware is not allowed for you",
  "getHardware" : "failure",
  "event" : "commandExecution"
}
```

getHardwareKey

Format: getHardwareKey()
Example: getHardwareKey()
Description: get a unique hardware key to create a license.

Response example:

```
{
  "event" : "commandExecution",
  "getHardwareKey" : "ok",
  "key" : "76F84C0-86ED79E-85EF743-801BF1B"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g.

onInviteSended.
- getHardwareKey – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- key – a text string which includes requested key.

Possible execution errors:

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getHardwareKey" : "failure"
}
```

- An attempt to run the command when you do not have permission:

```
{
  "error" : "getHardwareKey is not allowed for you",
  "getHardwareKey" : "failure",
  "event" : "commandExecution"
}
```

getMaxConfTitleLength

Format: getMaxConfTitleLength()
Example: getMaxConfTitleLength()
Description: get maximum length of the conference title.

Response example:

```
{
  "event" : "commandExecution",
  "getMaxConfTitleLength" : "ok",
  "length" : 255
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getMaxConfTitleLength – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- length – maximum length of the conference title.

Possible execution results:

- An attempt to run the conference with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
```

```
  "getMaxConfTitleLength" : "failure"
}
```

- An attempt to run the conference without permission:

```
{
  "error" : "getMaxConfTitleLength is not allowed for you",
  "getMaxConfTitleLength" : "failure",
  "event" : "commandExecution"
}
```

getMicMute

Format: getMicMute()
Example: getMicMute()
Description: to get the information on the microphone state (turned on or turned off).

Response example:

```
{
  "event" : "commandExecution",
  "getMicMute" : "ok",
  "mute" : false
}
```

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getMicMute – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- mute – indicates if the microphone is muted.

Possible execution errors:

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getMicMute" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getMicMute is not allowed for you",
  "getMicMute" : "failure",
  "event" : "commandExecution"
}
```

getMonitorsInfo

Format: getMonitorsInfo()
Example: getMonitorsInfo()
Description: get the information about monitors .

Response example:

```
{
   "getMonitorsInfo": "ok",
   "monitors": [
      {
         "name": "LG FULLHD(Analog) (NVIDIA GeForce 210 )",
         "primary": true,
         "x": 0,
         "y": 0,
         "width": 1920,
         "height": 1080,
         "index": 0
      },
      {
         "name": "Generic PnP Monitor (Intel(R) HD Graphics 4600)",
         "primary": false,
         "x": 1920,
         "y": 0,
         "width": 1024,
         "height": 768,
         "index": 1
      }
   ],
   "event": "commandExecution"
}
```

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getMonitorsInfo – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- name – monitor name (the driver in use is specified in parenthesis).
- primary – indicates if the monitor is primary.
- x – x-coordinate.
- y – y-coordinate.
- width – width.
- height –height.
- index – monitor index.

Possible execution errors
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getMonitorsInfo" : "failure"
}
```

getIdListInviteInConference

Format: getIdListInviteInConference()
Example: getIdListInviteInConference()
Description: get the list of the users requesting authorization to enter the conference.

Response example:

```
{
  "event" : "commandExecution",
  "getIdListInviteInConference" : "ok",
  "peerIdList" : [
    {
      "callId" : "ivanov@trueconf.com",
      "peerDn" : "Ivan Ivanov"
    },
    {
      "callId" : "petrov@trueconf.com",
      "peerDn" : "1"
    }
  ]
}
```

Parameter description

- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getIdListInviteInConference is a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- peerIdList – the list of the users requesting permission to enter the conference.
- callId – unique user ID.
- peerDn – display username which is encoded in BASE64 sequence. It needs reverse decoding.

Possible execution errors

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getIdListInviteInConference" : "failure"
}
```

- An attempt to run the command without permission:


```
{
  "error" : "getIdListInviteInConference is not allowed for you",
  "getIdListInviteInConference" : "failure",
  "event" : "commandExecution"
}
```


- An attempt to run the command when you are not in the conference:


```
{
  "error" : you're not in conference",
  "getIdListInviteInConference" : "failure",
  "event" : "commandExecution"
}
```


- An attempt to run the command when you are not in a group conference:


```
{
  "error" : "you're not in group conference",
  "getIdListInviteInConference" : "failure",
  "event" : "commandExecution"
}
```


- An attempt to run the command when you are not the conference owner:


```
{
  "error" : "you're not conference owner",
  "getIdListInviteInConference" : "failure",
  "event" : "commandExecution"
}
```


getModes

Format: getModes("base64_encodedDeviceName")
Example: getModes("base64_encode(AVerMedia HD Capture)")
Description: get the list of modes and pins for the specified capture board. The command is run immediately and the result of execution is received at once.

Response example:


```
{
  "getModes" : "ok",
  "activemode" : "NTSC_433",
  "activepin" : "RGB",
  "event" : "commandExecution",
  "modeList" : [
    "NTSC_433",
    "NTSC_M",
    "NTSC_M_J",
```

```
    "PAL_60",
    "PAL_B",
    "PAL_D",
    "PAL_H",
    "PAL_I",
    "PAL_M",
    "PAL_N",
    "SECAM_B",
    "SECAM_D",
    "SECAM_G",
    "SECAM_H",
    "SECAM_K",
    "SECAM_K1",
    "SECAM_L",
    "SECAM_L1"
  ],
  "pinList" : [ "RGB", "Serial Digital" ]
}
```

Input parameters description:
- devicename – device name.

Return parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getModes – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- activemode – device active mode.
- activepin – device active pin.
- modeList – the list of modes available for the device.
- pinList – the list of pins available for the device.
- supportPTZ – support for ptz control.

Possible execution errors:

- An attempt to run the command without parameters;

```
{
  "error" : "parameters must be not empty",
  "event" : "commandExecution",
  "getModes" : "failure"
}
```

- An attempt to run the command without permission;

```
{
  "error" : "getModes is not allowed for you",
  "getModes" : "failure",
  "event" : "commandExecution"
}
```

getSettings

Format: getSettings()
Example: getSettings()
Description: get the settings list. The command is run immediately and the result of execution is received at once.

Response example:


```
{
{
   "aecEnable": false,
   "audioPlayLevel": 0.74,
   "audioPlayerEnable": true,
   "audioRecordEnable": true,
   "audioRecordLevel": 0.66,
   "autoAccept": false,
   "autoAcceptPodiumFromOwner": false,
   "autoAllowPartToTakePodium": false,
   "defaultMultiConfMatrix": 0,
   "defaultP2PMatrix": 4,
   "enableAutologin": true,
   "inputBandWidth": 512,
   "language": 1,
   "lastSelectedConference": 0,
   "outputBandWidth": 256,
   "selfViewMirror": true,
   "videoEnableByUser": true,
   "getSettings": "ok",
   "event": "commandExecution"
}
}
```


Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getSettings – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
    - aecEnable – Echo cancellation. It can be true or false
    - audioPlayLevel – Audio playback level. Range of values: 0.00 - 1.0
    - audioRecordLevel – Audio capture level. Range of values: 0.00 - 1.0
    - autoAllowPartToTakePodium – allow the participants to take the podium automatically when you are the role-based conference owner. It can be true or false
    - autoAcceptPodiumFromOwner – automatically accept invitation to take

the podium when you are invited to take the podium in a role-based conference. It can be true or false
- autoAccept – automatically accept calls and invitations. It can be true or false
- audioRecordEnable – the parameter which shows if the recording is enabled.
- audioPlayerEnable – the parameter which shows if the audio playback is enabled.
- videoEnableByUser – the parameter which shows if the camera is muted.
- selfViewMirror – Self-view mirrored image. It can be true or false
- enableAutologin – enable auto login. It can be true or false
- inputBandWidth – Input bandwidth. It can be 32, 64, 128, 256, 512, 1024, 2048
- language – localization. It can be
    * 1 – Russian locale,
    * 2 – Polish locale,
    * 3 - others

- defaultP2PMatrix – default video layout for peer-to-peer video call.
- defaultMultiConfMatrix – default video layout for multiuser conference.
    * 0 – all the videos are the same (it is set by default for multiuser conferences).
    * 1 – one video is big while others are small.
    * 2 – one video
    * 3 – reserved and is not used as a video layout by default.
    * 4 – user's video plus your own video in the corner (it is set by default in peer-to-peer video call).
- lastSelectedConference – the latest conference type you selected. It can be
    * symmetric = 0 (symmetric)
    * asymmetric = 1 (asymmetric)
    * role = 2 (role-based)
- outputBandWidth – output bandwidth. It can be 32, 64, 128, 256, 512, 1024, 2048

Possible execution errors:
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getSettings" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getSettings is not allowed for you",
  "getSettings" : "failure",
  "event" : "commandExecution"
}
```

getSystemInfo

Format getSystemInfo()
Example: getSystemInfo()
Description: get the system information.

Response example:


```
{
   "authInfo": {
      "login": "petrov@trueconf.com",
      "displayName": "Ivan Ivanov 2"
   },
   "productInfo": {
      "major": 2,
      "minor": 0,
      "revision": 0,
      "build": 308
   },
   "fileInfo": {
      "major": 2,
      "minor": 0,
      "revision": 0,
      "build": 308
   },
   "serverInfo": {
      "server_ip": "91.109.204.28",
      "server": "ru7.trueconf.net#as",
      "service": true,
      "port": 443
   },
   "permissionsInfo": {
      "tariffName": "Free",
      "p2p": 1,
      "commonMulti": 1,
      "needPassword": 1,
      "createMulti": 1,
      "symMaxNumber": 3,
      "asymMaxNumber": 0,
      "roleMaxNumber": 0,
      "rlMaxNumber": 0,
      "canUseSlideShow": 1,
      "canUseDesktopSharing": 1,
      "canChangeAddressBook": 1,
      "canEditGroups": 1,
      "canUseDialer": 0
   },
   "cpuRating": {
      "benchRate": 227826,
      "rcvRate": 61,
```

```
      "sndRate": 56
   },
   "bitrateLimits": {
      "min": 32,
      "max": 4096
   },
   "captureFormat": {
      "width": 1280,
      "height": 720,
      "fps": 1500,
      "format" : 1
   },
   "getSystemInfo": "ok",
   "event": "commandExecution"
}
```


Return parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- authInfo – the parameter which is available only if your terminal is logged in. It is the set of parameters which are received after authorization.
- serverInfo – Server information.
- productInfo – Product information.
- bitrateLimits – Bitrate information.
- cpuRating – CPU information.
- fileInfo – File information.
- captureInfo – The information about the captured video resolution and mode.
- width – captured video width.
- height – captured video height.
- fps – the number of frames per second.
- format – video format. It can be:
   * YUYV = 0,
   * YUY2 = 1,
   * YVYU = 2,
   * MJPG = 3,
   * I420 = 4,
   * IYUV = 5,
   * UYVY = 6,
   * HDYC = 7,
   * YV12 = 8,
   * NV12 = 9,
   * NV16 = 10,
   * NV21 = 11,
   * RGB32 = 12,
   * RGB24 = 13,
   * ARGB = 14,
   * BGRA = 15,

  * YUV444 = 16,
  * H264 = 17,
  * H264_ES = 18,
  * H265 = 19,
  * VP80 = 20,
  * VP90 = 21,
  * STR0 = 22,
  * I420_STR0 = 23,

- permissionsInfo – Permission information.
- service – the parameter which shows if the client is connected to the service or to the server. It can be true or false. If the client is connected to the server, server box will contain IP address. Otherwise, it will contain service name.
- major, minor, revision, build – version information.
- peerDn – display name of the user (you) who has been logged in. It is encoded in BASE64 sequence and needs reverse decoding.
- getSystemInfo – command result. It can be ok, if successful, or failure, in case of a fault. In case of a fault you will see the error box containing error description.
- login – your unique user ID (*TrueConfID*)
- port – the port of the server you have been connected to.
- server – the name of the server you have been connected to.
- server_ip – IP of the server you have been connected to.
- tariffName – Your tariff plan. It can be
  * Free
  * Maximum
  * Corporate
- benchRate – general performance rate
- rcvRate – a value which characterizes performance rate when decoding video.
- sndRate – a value which characterizes performance rate when encoding video.
- max – maximum value.
- min – minimal video.
- asymMaxNumber – maximum number of asymmetric conference participants (if 0, you are not allowed to start this conference type).
- call – a parameter which specifies if a user can start a video call.
- commonMulti – a parameter which specifies if an account can start a multiuser conference.
- createMulti – a parameter which specifies if a multiuser conference is available to be started.
- needPassword - a parameter which specifies if you need a password in a multiuser conference.
- rlMaxNumber - maximum number of participants who can take the podium at the same time (if 0 , you are not allowed to start this conference type).
- roleMaxNumber – maximum number of role-based conference participants (if 0 , you are not allowed to start this conference type).
- symMaxNumber – maximum number of symmetric conference participants (if 0 , you are not allowed to start this conference type).

- canUseSlideShow – a parameter which indicates if you can use slide show.
- canUseDesktopSharing – a parameter which indicates if desktop sharing is available.
- canChangeAddressBook – a parameter which specifies if you can change address book.
- canEditGroups – a parameter which specifies of you can edit groups.
- canUseDialer – a parameter which specifies if dialer is available.

Possible execution errors:

- An attempt to run the command with pre-filled parameters.

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getSystemInfo" : "failure"
}
```

- An attempt to run the command without permission.

```
{
  "error" : "getSystemInfo is not allowed for you",
  "getSystemInfo" : "failure",
  "event" : "commandExecution"
}
```

getUserAvatar

Format: getUserAvatar("base64_encodedJsonParams")
Description: get user's avatar
Request example: getUserAvatar(base64_encodedJsonParams),
Example base64_encodeJsonParams:

```
{
  "peerId" : "ivanov@trueconf.com"
}
```

Parameter description
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getUserAvatar – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- peerId - callId of the user whose avatar you need to get.
- avatar - base64 image source.

Response example:

```
{
  "event" : "commandExecution",
  "getUserAvatar" : "ok",
  "avatar" : "base64 src"
```

```
}
```

Possible execution errors:

- required "peerId" parameter is not available:

```
{
  "error" : "there is no 'peerId' parameter",
  "getUserAvatar" : "failure",
  "event" : "commandExecution"
}
```

- Incorrect JSON format:

```
{
  "error" : "Bad JSON format",
  "getUserAvatar" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getUserAvatar is not allowed for you",
  "getUserAvatar" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you are not logged in:

```
{
  "getUserAvatar": "failure",
  "error": "you must be logged in to run this command",
  "event": "commandExecution"
}
```

getVideoMatrix

Format: getVideoMatrix()
Example: getVideoMatrix()
Description: get the information about current video matrix.

Response example:

```
{
  "event" : "commandExecution",
  "getVideoMatrix" : "commandExecution",
  "mainWindowHeight" : 544,
  "mainWindowWidth" : 960,
  "matrixType" : 4,
  "participants" : [
    {
      "height" : 181,
      "left" : 0,
      "peerId" : "test2@ub0n3.trueconf.name",
```

```
        "priority" : 0,
        "slotId" : 0,
        "top" : 363,
        "width" : 319
      }
      {
        "height" : 544,
        "left" : 0,
        "peerId" : "test1@ub0n3.trueconf.name",
        "priority" : 1,
        "slotId" : 1,
        "top" : 0,
        "width" : 960
      }
    }
}
```


Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getVideoMatrix – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- peerId – unique user ID TrueConfID
- participants – the list of video slots and conference participants.
- slotId – slot ID.
- top – top coordinates.
- width – window width.
- height – window height.
- mainWindowWidth – main window width.
- mainWindowHeight – main window height.
- left – left coordinates.
- priority – priority. It can be:
    * PRIORITY_LOW = 0
    * PRIORITY_HIGH = 1
- matrixType – layout matrix type. Matrix can be the following:
    * even= 0, all the windows are the same (for multiuser conference)
    * big = 1, one window is big while others are small (for multiuser conference);
    * one = 2, show only one window: the first video in the list of participants (for any conference type);
    * p2p = 3, reserved;
    * oneSelf= 4, a big user's window and a small self-view in the corner of the screen (for peer-to-peer video call);
    * PSshow = 5 show slide show, user and a self-view (for peer-to-peer video call);

Possible execution errors:

- An attempt to run the command without permission:

```
{
  "error" : "getRestrictions is not allowed for you",
  "getRestrictions" : "failure",
  "event" : "commandExecution"
}
```

getVideoMute

Format: getVideoMute()
Example: getVideoMute()
Description: get current status of video streaming.

Response example:

```
{
  "event" : "commandExecution",
  "getVideoMute" : "ok",
  "mute" : true
}
```

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getVideoMute – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- mute – a flag indicating of streaming is enabled.

Possible execution errors:

- An attempt to run the command with parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "getVideoMute" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "getVideoMute is not allowed for you",
  "getVideoMute" : "failure",
  "event" : "commandExecution"
}
```

getLicenseType

Format: getLicenseType()
Example: getLicenseType()
Description: get the information about pre-installed license.
Response example:

```
{
```

```
  "event" : "commandExecution",
  "getLicenseType" : "ok",
  "value" : 0
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- getLicenseType – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- value – installed license type (o is absent, 1 – base (part), 2 - full).

Note: the command is requested before the authorization, uid parameters and scope are not required when calling.

gotoPodium

Format: gotoPodium()
Example: gotoPodium()
Description: it is used to take the podium. If you are the conference owner, you will take it immediately, if not, you will have to wait until the conference owner allows you to take the podium. You will be informed about taking the podium with a respective notification (when onRoleEventOccurred notification is received). The command is run immediately and the result of execution is received at once.

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- gotoPodium – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:
```
{
  "event" : "commandExecution",
  "gotoPodium" : "ok"
}
```
Possible execution errors:
- An attempt to take the podium of the terminal is not in a role-based conference:

```
{
  "error" : "You're not in role conference",
  "event" : "commandExecution",
  "gotoPodium" : "failure"
}
```

- An attempt to take the podium if the terminal is out of the conference:

```
{
  "error" : "You're not in conference",
  "event" : "commandExecution",
  "gotoPodium" : "failure"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "gotoPodium" : "failure"
}
```

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "gotoPodium" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "gotoPodium is not allowed for you",
  "gotoPodium" : "failure",
  "event" : "commandExecution"
}
```

hangUp

Format: hangUp() or hangUp("base64_encoded(true|false)")
Example: hangUp(), hangUp("base64_encodedFalse")
Description: end a call or a conference. The command is used when the conference has already been created. hangUp() format is used during a video call. During group conferences both formats are used. By using hangUp() format you leave the conference, but other participants remain in the conference. By using hangUp("true", the conference ends for all the participants. hangUp("true") is used only if you are the conference owner, otherwise a failure occurs. Positive response (”ok”) means the command has been accepted for execution but has not been run executed yet. Execution result will be received separately via notification.

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- hangUp – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "hangUp" : "ok"
}
```

Possible execution errors:

- An attempt to end a video call or a video conference in the terminal state 3 (see **getAppState()**):

```
{
  "error" : "You're not in conference",
  "event" : "commandExecution",
  "hangUp" : "failure"
}
```

- An attempt to end a video call with pre-filled parameters:

```
{
  "error" : "you're in p2p conference, parameters must be empty",
  "event" : "commandExecution",
  "hangUp" : "failure"
}
```

- An attempt to end a call if the parameter is empty while the conference is ending:

```
{
  "error" : "you should specify whether conference should be terminated for all the participants",
  "event" : "commandExecution",
  "hangUp" : "failure"
}
```

- An attempt to end a call having specified the wrong parameter (it should be either true or false):

```
{
  "error" : "wrong hangUp parameter",
  "event" : "commandExecution",
  "hangUp" : "failure"
}
```

- An attempt to end a call with true parameter if you are not the conference owner:

```
{
  "error" : "you're not conference owner, wrong parameter",
  "event" : "commandExecution",
  "hangUp" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "hangUp is not allowed for you",
  "hangUp" : "failure",
  "event" : "commandExecution"
}
```

inviteToConference

Format: inviteToConference("base64PeerId")
Example: inviteToConference("base64_encoded(petrov@trueconf.com)")
Description: invite a user into the conference. It can be used only by the conference owner. The command is run immediately and the result of execution is received at once.

Parameter description:
- peerId – unique user ID (TrueConfID).
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- inviteToConference is a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

```
{
  "event" : "commandExecution",
  "inviteToConference" : "ok"
}
```

Possible execution errors:
- An attempt to invite a user into the conference during a video call:

```
{
  "error" : "You're in p2p conference",
  "event" : "commandExecution",
  "inviteToConference" : "failure"
}
```

- An attempt to invite a participant into a conference of 3 terminal state (getAppState()):

```
{
  "error" : "you're not in conference",
  "event" : "commandExecution",
  "inviteToConference" : "failure"
}
```

- An attempt to invite a user into the conference having entered an empty peerId:

```
{
  "error" : "peerId is empty",
```

```
  "event" : "commandExecution",
  "inviteToConference" : "failure"
}
```

- An attempt to invite into the conference having specified a unique user ID (TrueConfID) in the parameters:

```
{
  "error" : "you can't invite yourself",
  "inviteToConference" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to invite users into the conference if you are not the conference moderator:

```
{
  "error" : "you're not conference owner",
  "inviteToConference" : "failure",
  "event" : "commandExecution"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "inviteToConference" : "failure"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "inviteToconference" : "failure",
  "event" : "commandExecution"
}
```

- You have entered incomplete peerId:

```
{

  "error" : "peerId must be in a full form",

  "event" : "commandExecution",

  "inviteToConference" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "inviteToConference is not allowed for you",
  "inviteToConference" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to invite a participant into the conference when there are already maximum number of participants taking part in the conference:

```
{
  "error" : "maximum participants number in conference reached",
  "inviteToConference" : "failure",
  "event" : "commandExecution"
}
```

inviteToPodium

Format: inviteToPodium("base64PeerId")
Example: inviteToPodium("base64_encoded(petrov@trueconf.com)")
Description: invite a user to take the podium. The command is possible only in a group conference. The command is run immediately and the result of execution is received at once.

Parameter description:
- peerId – unique user ID (TrueConfID)
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- inviteToPodium – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "inviteToPodium" : "ok"
}
```

Possible execution errors:
– An attempt to invite a participant to take the podium if the terminal is out of the conference:

```
{
  "error" : "You're not in conference",
  "event" : "commandExecution",
  "inviteToPodium" : "failure"
}
```

– An attempt to invite a conference participant to take the podium if the conference is not a role-based one:

```
{
  "error" : "You're not in role conference",
  "event" : "commandExecution",
  "inviteToPodium" : "failure"
}
```

– An attempt to invite yourself to take the podium. In this case you need to run gotoPodium() command:

```
{
  "error" : "you should use gotoPodium() command",
  "event" : "commandExecution",
  "inviteToPodium" : "failure"
}
```

– An attempt to invite a participant to take the podium if you are not the conference owner:

```
{
  "error" : "you're not conference owner",
  "event" : "commandExecution",
  "inviteToPodium" : "failure"
}
```

- An attempt to invite a participant to take the podium if this participant has already taken the podium. Error message will contain the participant's peerId

```
{
  "error" : "peerId is already on the podium",
  "event" : "commandExecution",
  "inviteToPodium" : "failure"
}
```

- An attempt which occurs if the participant with the peerId specified in the parameters could not be found in the conference:

```
{
  "error" : "peerId not found",
  "event" : "commandExecution",
  "inviteToPodium" : "failure"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "inviteToPodium" : "failure"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "inviteToPodium" : "failure",
  "event" : "commandExecution"
}
```

- Incomplete peerId

```
{
```

```
  "error" : "peerId must be in a full form",
  "event" : "commandExecution",

  "inviteToPodium" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "inviteToPodium is not allowed for you",
  "inviteToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to invite a participant to take the podium when there is already maximum number of participants taking part in the conference:

```
{
  "error" : "maximum participants number on podium reached",
  "inviteToPodium" : "failure",
  "event" : "commandExecution"
}
```

kickFromPodium

Format: kickFromPodium("base64PeerId")
Example: kickFromPodium("base64_encode(ivanov@trueconf.com)")
Description: remove a conference participant from the tribune. The command is run immediately and the result of execution is received at once.

Parameter description:
- peerId – unique user ID (TrueConfID).
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- kickFromPodium is a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "kickFromPodium" : "ok"
}
```

Possible execution errors:
- An attempt to remove a conference participant from the podium during a video call:

```
{
  "error" : "You're not in role conference",
  "event" : "commandExecution",
```

```
  "kickFromPodium" : "failure"
}
```

- An attempt to remove a conference participant from the podium when the terminal is in state 3 (see getAppState()):

```
{
  "error" : "you're not in conference",
  "event" : "commandExecution",
  "kickFromPodium" : "failure"
}
```

- An attempt to remove a conference participant from the podium if you are not the conference:

```
{
  "error" : "you're not conference owner",
  "event" : "commandExecution",
  "kickFromPodium" : "failure"
}
```

- An attempt to remove a conference participant from the podium if the user with peerId specified has not been found:

```
{
  "error" : "peerId not found",
  "event" : "commandExecution",
  "kickFromPodium" : "failure"
}
```

- An attempt to remove yourself from the podium if you are the conference owner. In this case you need to run the leavePodium command:

```
{
  "error" : "you should use leavePodium() command",
  "event" : "commandExecution",
  "kickFromPodium" : "failure"
}
```

- An attempt to remove a conference participant from the podium if the participant is not taking the podium now:

```
{
  "error" : "peerId is not on the podium",
  "event" : "commandExecution",
  "kickFromPodium" : "failure"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "kickFromPodium" : "failure"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "kickFromPodium" : "failure",
  "event" : "commandExecution"
}
```

- Incomplete peerId:

```
{
  "error" : "peerId must be in a full form",
  "event" : "commandExecution",
  "kickFromPodium" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "kickFromPodium is not allowed for you",
  "kickFromPodium" : "failure",
  "event" : "commandExecution"
}
```

kickPeer

Format: kickPeer("base64PeerId")
Example: kickPeer("base64_encode(ivanov@trueconf.com)")
Description: remove a participant from the conference using participant's peerId. It is used **only** in video conferences and **only** by the conference owner. The command is run immediately and the result of execution is received at once.

Parameters description:
- peerId – unique user ID (TrueConfID)
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- kickPeer – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be displayed.

Response example:

```
{
  "event" : "commandExecution",
  "kickPeer" : "ok"
}
```

Possible execution errors:
- This error occurs when you try to remove a participant from the conference using peerId of the participant who is not participating in the current conference:

```
{
  "error" : "peerId not found",
  "event" : "commandExecution",
  "kickPeer" : "failure"
}
```

- This error occurs when you try to remove a participant from the conference if the terminal is not in the conference:

```
{
  "error" : "You're not in conference",
  "event" : "commandExecution",
  "kickPeer" : "failure"
}
```

- An attempt to remove a participant from the conference during a video call:

```
{
  "error" : "you're in p2p conference",
  "event" : "commandExecution",
  "kickPeer" : "failure"
}
```

- An attempt to remove a participant from the conference if you are not the conference owner:

```
{
  "error" : "you're not conference owner",
  "event" : "commandExecution",
  "kickPeer" : "failure"
}
```

- An attempt to remove yourself from the conference:

```
{
  "error" : "don't kick yourself",
  "event" : "commandExecution",
  "kickPeer" : "failure"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "kickPeer" : "failure"
}
```

- An attempt to run the command with no parameters specified:

```
{
```

```
  "error" : "parameters must be not empty",
  "kickPeer" : "failure",
  "event" : "commandExecution"
}
```

- Incomplete peerId:

```
{
  "error" : "peerId must be in a full form",
  "event" : "commandExecution",
  "kickPeer" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "kickPeer is not allowed for you",
  "kickPeer" : "failure",
  "event" : "commandExecution"
}
```

leavePodium

Format: leavePodium()
Example: leavePodium()
Description: Leave the podium. The command is run immediately and the result of execution is received at once.

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- leavePodium – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "leavePodium" : "ok"
}
```

Possible execution errors:
- An attempt to leave the podium when you are not in the role-based conference:

```
{
  "error" : "You're not in role conference",
  "event" : "commandExecution",
  "leavePodium" : "failure"
}
```

- An attempt to leave the podium if the terminal is out of the conference:

```
{
  "error" : "you're not in conference",
  "event" : "commandExecution",
  "leavePodium" : "failure"
}
```

- An attempt to leave the podium if the terminal is not taking the podium now:

```
{
  "error" : "you're not on the podium",
  "event" : "commandExecution",
  "leavePodium" : "failure"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "leavePodium" : "failure"
}
```

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "leavePodium" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "leavePodium is not allowed for you",
  "leavePodium" : "failure",
  "event" : "commandExecution"
}
```

login

Format: login("base64_encodedLogin", "base64_encodedPassword")
Example: login("base64_encode(ivanov)", "base64_encode(my_password)")
Description: authorize on the server. ok (positive response) means the command has been accepted for execution but has not been executed yet. The execution result will be received via notification.

Parameter description:
- peerId – unique user ID (TrueConfID).
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.

- login – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "login" : "ok"
}
```

Possible execution errors:
- An attempt to authorize on the server when the terminal has not been connected to the server:

```
{
  "error" : "you should connect to server first",
  "login" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "login" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "login is not allowed for you",
  "login" : "failure",
  "event" : "commandExecution"
}
```

logout

Format: logout()
Example: logout()
Description: logout on the server. Receiving ok means the command has been accepted for the execution but it has not been successfully run yet. The execution result will be received via notification.

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- logout – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example

```
{
  "event" : "commandExecution",
  "logout" : "ok"
}
```

Possible execution errors:

- An attempt to log out when the client has not been logged in:

```
{
  "error" : "you're not logged in",
  "event" : "commandExecution",
  "logout" : "failure"
}
```

- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "logout" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "logout is not allowed for you",
  "logout" : "failure",
  "event" : "commandExecution"
}
```

ptzRight

Format: ptzRight()
Example: ptzRight()
Description: Turn the camera to the right. The command is run immediately and the result of execution is received at once.

Parameter description:

- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- ptzRight – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "ptzRight" : "ok"
```

```
}
```

Possible execution errors:
- An attempt to run the command with pre-filled parameters.

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "ptzRight" : "failure"
}
```

- An attempt to send a command not to a PTZ camera.

```
{
  "error" : "you must choose ptz camera",
  "event" : "commandExecution",
  "ptzRight" : "failure"
}
```

ptzLeft

Format: ptzLeft()
Example: ptzLeft()
Description: Turn the camera to the left. The command is run immediately and the execution result is received at once.
Parameters description
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- ptzLeft – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "ptzLeft" : "ok"
}
```

Possible execution errors:
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "ptzLeft" : "failure"
}
```

- An attempt to send a command not to a PTZ camera:

```
{
  "error" : "you must choose ptz camera",
  "event" : "commandExecution",
  "ptzLeft" : "failure"
```

```
}
```

ptzUp

Format: ptzUp()
Example: ptzUp()
Description: Rotate the camera up. The command is run immediately and the result of execution is received at once.

Parameters description
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- ptzUp – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "ptzUp" : "ok"
}
```

Possible execution errors:
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "ptzUp" : "failure"
}
```

- An attempt to send a command not to a PTZ camera:

```
{
  "error" : "you must choose ptz camera",
  "event" : "commandExecution",
  "ptzUp" : "failure"
}
```

ptzDown

Format: ptzDown()
Example: ptzDown()
Description: Rotate the camera down. The command is run immediately and the result of execution is received at once.

Parameters description :
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.

- ptzDown – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "ptzDown" : "ok"
}
```

Possible execution errors:
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "ptzDown" : "failure"
}
```

- An attempt to send a command not to a PTZ camera:

```
{
  "error" : "you must choose ptz camera",
  "event" : "commandExecution",
  "ptzDown" : "failure"
}
```

ptzZoomInc

Format: ptzZoomInc()
Example: ptzZoomInc()
Description: Zoom in the image. The command is run immediately and the result of execution is received at once.

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- ptzZoomInc – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "ptzZoomInc" : "ok"
}
```

Possible execution errors:
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
```

```
  "event" : "commandExecution",
  "ptzZoomInc" : "failure"
}
```

- An attempt to send a command not to a PTZ camera:

```
{
  "error" : "you must choose ptz camera",
  "event" : "commandExecution",
  "ptzZoomInc" : "failure"
}
```

ptzZoomDec

Format ptzZoomDec()
Example ptzZoomDec()
Description: Zoom the image out. The command is run immediately and the result of execution is received at once.

Parameters description :
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- ptzZoomDec – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example

```
{
  "event" : "commandExecution",
  "ptzZoomDec" : "ok"
}
```

Possible execution errors:
- An attempt to run the command with pre-filled parameters:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "ptzZoomDec" : "failure"
}
```

- An attempt to send a command not to a PTZ camera:

```
{
  "error" : "you must choose ptz camera",
  "event" : "commandExecution",
  "ptzZoomDec" : "failure"
}
```

reject

Format: reject()
Example: reject()
Description: The command allows to reject incoming call or invitation to the conference. The command is run immediately and the result of execution is received at once.

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- reject – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:
```
{
  "event" : "commandExecution",
  "reject" : "ok"
}
```

Possible execution errors:
- An attempt to reject a call when no one is calling:
```
{
  "error" : "nobody calling you",
  "event" : "commandExecution",
  "reject" : "failure"
}
```

- An attempt to run the command in a conference. In this case you need to run rejectPeer("peerId"):
```
{
  "error" : "you should use 'rejectPeer()' in conference",
  "event" : "commandExecution",
  "reject" : "failure"
}
```

- Unexpected error. Try again:
```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "reject" : "failure"
}
```

- An attempt to run the command with pre-filled parameters:
```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "reject" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "reject is not allowed for you",
  "reject" : "failure",
  "event" : "commandExecution"
}
```

rejectPeer

Format: rejectPeer(base64PeerId)
Example: rejectPeer(base64_encode("ivanov@trueconf.com"))
Description: reject user's request to join **your** conference. The command is run immediately and the result of execution is received at once.

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- rejectPeer is a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example

```
{
  "event" : "commandExecution",
  "rejectPeer" : "ok"
}
```

Possible execution errors
- An attempt to reject a request to join the conference if the terminal is not in the conference now:

```
{
  "error" : "you're not in conference",
  "event" : "commandExecution",
  "rejectPeer" : "failure"
}
```

- An attempt to reject a request to join the conference if you are not the conference owner:

```
{
  "error" : "you're not conference owner",
  "event" : "commandExecution",
  "rejectPeer" : "failure"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
```

```
  "event" : "commandExecution",
  "rejectPeer" : "failure"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "event" : "commandExecution",
  "rejectPeer" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "rejectPeer is not allowed for you",
  "rejectPeer" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command having specified callId of the user who is not calling you in the parameters:

```
{
  "error" : "peerId is not found in request invite list",
  "event" : "commandExecution",
  "rejectPeer" : "failure"
}
```

- An attempt to run the command having specified incomplete callId in the parameters:

```
{
   "rejectPeer": "failure",
   "error": "peerId must be in a full form",
   "event": "commandExecution"
}
```

rejectInvitationToPodium

Format: rejectInvitationToPodium()
Example: rejectInvitationToPodium()
Description: Reject the invitation to take the podium. The command is run immediately and the result of execution is received at once.

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- rejectInvitationToPodium – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:


```
{
  "event" : "commandExecution",
  "rejectInvitationToPodium" : "ok"
}
```


Possible execution errors:

- An attempt to reject the invitation to take the podium if the terminal is not in the conference:


```
{
  "error" : "you're not in conference",
  "rejectInvitationToPodium" : "failure",
  "event" : "commandExecution"
}
```


- An attempt to reject the invitation to take the podium if the terminal is not in a role-based conference:


```
{
  "error" : "you're not in role conference",
  "rejectInvitationToPodium" : "failure",
  "event" : "commandExecution"
}
```


- An attempt to reject the invitation to take the podium if you are the conference owner. In this case you do not need an invitation, you can easily run gotoPodium() and leavePodium() commands:


```
{
  "error" : "you're conference owner, you can't do this",
  "event" : "commandExecution",
  "rejectInvitationToPodium" : "failure"
}
```


- Unexpected error. Try again:


```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "rejectInvitationToPodium" : "failure"
}
```


- An attempt to run the command with pre-filled parameters:


```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "rejectInvitationToPodium" : "failure"
}
```


- An attempt to run the command without permission:


```
{
```

```
  "error" : "rejectInvitationToPodium is not allowed for you",
  "rejectInvitationToPodium" : "failure",
  "event" : "commandExecution"
}
```

rejectRequestToPodium

Format: rejectRequestToPodium()
Example: rejectRequestToPodium()
Description: Reject user's request to take the podium. The command is run immediately and the result of execution is received at once.

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- rejectRequestToPodium – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:
```
{
  "rejectRequestToPodium" : "ok",
  "event" : "commandExecution"
}
```

Possible execution errors:
- An attempt to run the command if the terminal is not in the conference now:
```
{
  "error" : "you're not in conference",
  "rejectRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```
- An attempt to run the command if the conference is not a role-based one:
```
{
  "error" : "you're not in role conference",
  "rejectRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you not the conference owner:
```
{
  "error" : "you're  not conference owner",
  "rejectRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command with parameters specified:

```
{
  "rejectRequestToPodium" : "failure",
  "error" : "parameters must be empty",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you do not have permission:

```
{
  "error" : "rejectRequestToPodium is not allowed for you",
  "rejectRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when no corresponding requests have been made:

```
{
  "error" : "there is no any requests",
  "rejectRequestToPodium" : "failure",
  "event" : "commandExecution"
}
```

removeContactFromAbook

Format: removeContactFromAbook(base64EncodedCallId)
Description: Remove a user from the Address Book. The command is run immediately and the result of execution is received at once.

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- removeContactFromAbook – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- error – present only if there was a mistake and it contains its description.
- callId – a callId of the user you need to remove from the Address Book in BASE64.

Response example:

```
{
  "event" : "commandExecution",
  "removeContactFromAbook" : "ok"
}
```

Possible execution errors:
- An attempt to run the command removeContactFromAbook() with no parameters specified.

```
{
  "error" : "parameters must be not empty",
```

```
  "event" : "commandExecution",
  "removeContactFromAbook" : "failure"
}
```

- An attempt to run the command with incorrect parameters specified

```
{
  "error" : "check parameters",
  "removeContactFromAbook" : "failure",
  "event" : "commandExecution"
}
```

setAudioMute

Format: setAudioMute("base64_encodedMute")
Example: setAudioMute("base64_encode(true)")
Description: turn on/off audio playback.

Parameter description:
- base64_encodedMute – parameter which indicates if it is turned on / off.
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended. –
setAudioMute – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "setAudioMute" : "ok"
}
```

Possible execution errors
- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "setAudioMute" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "setAudioMute is not allowed for you",
  "setAudioMute" : "failure",
  "event" : "commandExecution"
}
```

- Incorrect parameter:

```
{
```

```
  "error" : "bad param",
  "setAudioMute" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when no device has been selected:

```
{
  "error" : "error", "can't use this command with 'none' device",
  "setAudioMute" : "failure",
  "event" : "commandExecution"
}
```

setAuthParams

Format: setAuthParams("base64_encode(authParamsInJSON)")
Description: set a new password for managing the terminal by a user or administrator. The command is run immediately and the result of execution is received at once.

Example:
setAuthParams("base64_encode(authParamsInJSON)"),
where authParamsInJSON can be

```
{
   "type": "admin",
   "password": "password"
}
```

Input parameters description:
- type – a parameter which specifies the account for which you set a new password and username. Admin and User.
- password – a new password.

Response example:

```
{
  "setAuthParams" : "ok",
  "key" : "dsfsdfsdfdsfsdfsdfsdfsdf",
  "event" : "commandExecution"
}
```

Return parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- setAuthParams – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- key – a new key for notifications decoding.

Possible execution errors:

- Wrong JSON format

```
{
  "error" : "Bad JSON format",
  "setAuthParams" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command without type parameter which is required to indicate the account type:

```
{
  "error" : "parameter 'type' must be present",
  "setAuthParams" : "failure",
  "event" : "commandExecution"
}
```

- "type" parameter is not string type:

```
{
  "error" : "parameter 'type' must be string type",
  "setAuthParams" : "failure",
  "event" : "commandExecution"
}
```

- Wrong meaning of type parameter. It can be either admin, or user:

```
{
  "error" : "wrong 'type' parameter",
  "setAuthParams" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when one of the required fields is absent.

```
{
  "error" : "parameter 'password' must be present",
  "setAuthParams" : "failure",
  "event" : "commandExecution"
}
```

- Wrong 'password' parameter type:

```
{
  "error" : "parameter 'password' must be string type",
  "setAuthParams" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to change administrator password when you are logged in as a user:

```
{
   "setAuthParams": "failure",
   "error": "you can change password only for 'user' type",
   "event": "commandExecution"
```

```
}
```

setBroadcastSelfie

Format: setBroadcastSelfie("base64_encode(paramsInJSON)")
Description: turn on to view the images from the current video signal source. The command is run immediately and the result of execution is received at once. The frames will be received from callId of the user you are logged in as. If the terminal has not logged in, the frames will be received from **VideoCaptureSlot** callId

Example: setBroadcastSelfie("base64_encode(paramsInJSON)"), where paramsInJSON can be, for example

```
{
   "enabled": true,
   "fps": 3
}
```

Input parameters description:
- enabled is the parameter which specifies if you need to turn the video display on or off. It can be true and false.
- fps is an integer-valued parameter which specifies the number of sent frames per second. It has the value [1..N].

Response example:

```
{
  "setBroadcastSelfie" : "ok"
}
```

Return parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- setBroadcastSelfie – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Possible execution errors:
- Wrong JSON format:

```
{
  "error" : "Bad JSON format",
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you do not have an active video signal source:

```
{
  "error" : "there is no active camera",
```

```
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when the required enabled parameter is not specified:

```
{
  "error" : "there is no 'enabled' parameter",
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to specify the value for enabled parameter not as a bool type:

```
{
  "error" : "'enabled' parameter in not bool",
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you do not have a required fps parameter:

```
{
  "error" : "there is no 'fps' parameter",
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to indicate the value for enabled parameter not as int type:

```
{
  "error" : "'fps' parameter in not int",
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when fps parameter is wrong

```
{
  "error" : "'fps' parameter is bad",
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you do not have permission:

```
{
  "error" : "setBroadcastSelfie is not allowed for you",
  "setBroadcastSelfie" : "failure",
  "event" : "commandExecution"
}
```

setHardware

Format: setHardware("base64_encodedHardwareInJson")
Description: set current audio / video capture and playback devices. Parameters specify name of each device which has been been received by running getHardware() command. If you specify the device which does not exist, an error will occur. The command is run immediately and the result of execution is received at once.
Example: setHardware("base64_encode(hardwareInJson)"),
where hardwareInJson can be for example:

```
{
   "videoCapture": "HP Webcam",
   "audioCapture": "none",
   "audioPlayback": "badValue",
   "badKey": "none"
}
```

Input parameter description: :
- videoCapture – a parameter specified to set video capture device. Besides enlisted possible values, returned by getHardware() command, you can also use none (to turn off the device).
- audioCapture – a parameter specified to set audio capture device. Besides enlisted possible values, returned by getHardware() command, you can also use none (to turn off the device).
- audioPlayback – a parameter specified to set audio playback device. Besides enlisted possible values, returned by getHardware() command, you can also use none (to turn off the device).

Response example:

```
{
  "audioCapture" : "ok",
  "audioPlayback" : "failure",
  "badKey" : "Failure: no such parameter",
  "errorAudioPlayback" : "No such audioPlayback",
  "videoCapture" : "ok",
  "event" : "commandExecution",
  "setHardware" : "ok"
}
```

Return parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- setHardware – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.
- audioPlayback is a result of turning off the audio playback device. It can be:
    * ok – the device has been successfully assigned.

    * failure – the device has not been successfully assigned. In this case ErrorAudioPlayback with error description will be displayed.
- audioCapture – a result of audio capture device setting. It can be
* ok – the device has been successfully assigned.
    * failure – the device has not been successfully assigned. In this case ErrorAudioPlayback with error description will be displayed.
- videoCapture – a result of video capture device setting. It can be
    * ok – the device has been successfully assigned.
    * failure – the device has not been successfully assigned. In this case ErrorAudioPlayback with error description will be displayed.
- When you try to enter a wrong key, for example, “BadKey”, an error with description will be displayed. For example: "BadKey" : "Failure: no such parameter".

Possible execution errors:
- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "setHardware" : "failure",
  "event" : "commandExecution"
}
```

- You have specified "AudioCapture" not as a string parameter:

```
{
  "error" : "audioCapture should be string",
  "setHardware" : "failure",
  "event" : "commandExecution"
}
```

- You have specified "audioPlayback" not as a string parameter:

```
{
  "error" : "audioPlayback should be string",
  "setHardware" : "failure",
  "event" : "commandExecution"
}
```

- You have specified "VideoCapture" not as a string parameter:

```
{
  "error" : "videoCapture should be string",
  "setHardware" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "setHardware is not allowed for you",
  "setHardware" : "failure",
  "event" : "commandExecution"
}
```

setMicMute

Format: setMicMute("base64_encodedMute")
Example: setMicMute("base64_encode(true)")
Description: mute / unmute microphone

Parameters description:
- base64_encodedMute – a parameter indicating if it is muted or unmuted.
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- setMicMute – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:

```
{
  "event" : "commandExecution",
  "setMicMute" : "ok"
}
```

Possible execution errors:
- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "setMicMute" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "setMicMute is not allowed for you",
  "setMicMute" : "failure",
  "event" : "commandExecution"
}
```

- Wrong parameter:

```
{
  "error" : "bad param",
  "setMicMute" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when no device has been selected:

```
{
  "error" : "error", "can't use this command with 'none' device",
  "setMicMute" : "failure",
```

```
  "event" : "commandExecution"
}
```

setModes

Format: setModes("base64_encodedPin", "base64_encodedMode")
Description: set a pin and video capture card mode. Parameters indicate pin value and active device mode value. If you indicate non-existing parameter values, an error occurs. The command is run immediately and the result of execution is received at once.
Example: setModes("base64_encode(Serial Digital)","base64_encode(SECAM_B)")

Input parameters description:
- pin – a parameter setting a pin for an active video capture device. It should be included into pinList, returned by getModes() command.
- mode – a parameter which sets pin value for active video capture device. It should be included into modeList, returned by getModes() command.

Response example:

```
{
  "setModes" : "ok",
  "event" : "commandExecution"
}
```

Return parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- SetMode – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Possible execution errors:
- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "setModes" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to indicate “Pin” or “mode” parameters which are not included into permitted parameters:

```
{
  "error" : "Wrong pin or mode value",
  "setModes" : "failiture",
  "event" : "commandExecution"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "setModes is not allowed for you",
  "setModes" : "failure",
  "event" : "commandExecution"
}
```

setSettings

Format: setSettings("base64_encodedSettingsInJson")
Description: set application settings. Possible settings list and description can be found here. The command is run immediately and the result of execution is received at once.

Example: setSettings("base64_encode(settingsInJson)"),
where settingsInJson – is a parameter in JSON format, for example:

```
{
   "audioPlayLevel": 0.55,
   "enableAutologin": true
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- setSettings – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:
The list containing settings name and setting result. It can be ok if the settings have been applied, or not found, if setting name has not been found:

```
{
  "audioPlayLevVel" : "not found",
  "enableAutologin" : "ok",
  "event" : "commandExecution",
  "setSettings" : "ok"
}
```

- An attempt to run the command with no parameters specified:

```
{
  "error" : "parameters must be not empty",
  "setSettings" : "failure",
  "event" : "commandExecution"
}
```

setVideoMute

Format: setVideoMute("base64_encodedMute")
Example: setVideoMute("base64_encode(true)")
Description: turn video streaming on and off.

Parameter description:
- base64_encodedMute – a parameter indicating if the streaming has been turned on or off.
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- setVideoMute – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:
```
{
  "event" : "commandExecution",
  "setVideoMute" : "ok"
}
```

Possible execution errors:
- An attempt to run the command with no parameters specified:
```
{
  "error" : "parameters must be not empty",
  "setVideoMute" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when you do not have permission:
```
{
  "error" : "setAudioMute is not allowed for you",
  "setVideoMute" : "failure",
  "event" : "commandExecution"
}
```

- Wrong parameter:
```
{
  "error" : "bad param",
  "setVideoMute" : "failure",
  "event" : "commandExecution"
}
```

- An attempt to run the command when the device has not been selected:
```
{
  "error" : "error", "can't use this command with 'none' device",
  "setVideoMute" : "failure",
```

```
  "event" : "commandExecution"
}
```

startPictureBroadcast

Format: startPictureBroadcast()
Example: startPictureBroadcast()
Description: Start broadcasting of the pre-downloaded image instead of your video. Before the function call, you need to download the image via POST or PUT HTTP-request. The request should be sent to the address where web interface is loaded. It should be sent to the port for built-in http-image server (see embeddedHttpPort parameter in the application state received by getAppState function.

startRemark

Format: startRemark()
Example: startRemark()
Description: make an audio remark in a role-based conference. If the remark is busy, an error occurs. After positive response you will have 5 seconds to make a remark. After each new request remark time is reset again for 5 seconds. With each second passing you will get a notification about the time left (onRemarkCountDown). The command is run immediately and the result of execution is received at once.

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- startRemark – a command execution result. It can be ok if successful, or failure in case of failure. If it is failure an error field with error description will be shown.

Response example:
```
{
  "event" : "commandExecution",
  "startRemark" : "ok"
}
```
Possible execution errors:
- An attempt to make an audio remark if another participant is using the remark right now:
```
{
  "error" : "another participant use remark now",
  "event" : "commandExecution",
  "startRemark" : "failure"
}
```

- An attempt to make an audio remark is the terminal is not in the conference right now:

```
{
  "error" : "you're not in conference",
  "event" : "commandExecution",
  "startRemark" : "failure"
}
```

- An attempt to make a remark if you are not in a role-based conference:

```
{
  "error" : "you're not in role conference",
  "event" : "commandExecution",
  "startRemark" : "failure"
}
```

- An attempt to make an audio remark is you are the conference owner. To do this, you need to run gotoPodium command:

```
{
  "error" : "you're conference owner, use gotoPodium command",
  "event" : "commandExecution",
  "startRemark" : "failure"
}
```

- Unexpected error. Try again:

```
{
  "error" : "something went wrong, try again",
  "event" : "commandExecution",
  "startRemark" : "failure"
}
```

- An attempt to run the command with parameters specified:

```
{
  "error" : "parameters must be empty",
  "event" : "commandExecution",
  "startRemark" : "failure"
}
```

- An attempt to run the command without permission:

```
{
  "error" : "startRemark is not allowed for you",
  "startRemark" : "failure",
  "event" : "commandExecution"
}
```

stopPictureBroadcast

Format: stopPictureBroadcast()
Example: stopPictureBroadcast()
Description: Stop startPictureBroadcast broadcasting of the pre-downloaded image instead of video signal.

Notifications

onAbookUpdate

Description: a notification you receive after Address Book update.
Example:

```
{
  "abook" : [
    {
      "peerId" : "demo128@trueconf.com",
      "peerDn" : "128Kb demo",
      "status" : 0
    }
  ],
  "event" : "onAbookUpdate"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – a unique user ID TrueConfID
- peerDn – user display name which is encoded in BASE64 sequence. It needs reverse decoding
- status – user's status which can have one of the following values:
  * USER_INVALID = -1,
  * USER_LOGOFF = 0,
  * USER_AVAIL = 1,
  * USER_BUSY = 2,
  * USER_MULTIHOST = 5

onAppUpdateAvailable

Description: a notification you receive when client application update is available.
Example:

```
{
  "event" : "onAppUpdateAvailable",
  "version" : "2.0.0",
  "mustUpdate": false
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- version – client application version available for an update.

- mustUpdate – a parameter which indicates if the application is required (true if the current app version is lower than minimal supported version.

onAuthorizationNeeded

Description: a notification you receive after the terminal switched off remote management clients using GUI menu. Practically, updates encryption key has been changed, while authorized clients list has been removed. In this case you need to pass an authorization process once again. All future updates will be encrypted with another key.
Example:

```
{
  "event" : "onAuthorizationNeeded",
  "type" : 0,
  "excludeCid" :
"90f3b7f99d34eb401f774d16284b92fcd10e815c9fd710f6fc61b599a980e129
"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- version – client application version available for an update.
- type – indicates a user who needs to be re-authorized. It can be
  * 0 = Admin
  * 1 = User
- excludeCid – optional indicator. It means cid of the client which does not need to be authorized again. You get cid during authorization process.

onChangeVideoMatrixReport

Description: a notification which includes a change video matrix report.
Example:

```
{
  "event" : "onChangeVideoMatrixReport",
  "result" : false,
  "error" : "error description"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- result – a command result. It can be true or false
- error – optional indicator. It includes error description.

onConferenceCreated

Description: a notification you receive when
* a video call is started
* a group conference is started

Example:

```
{
  "confName" : "Monthly report",
  "confType" : 1,
  "confId" : "00042f28@ru9.trueconf.net#as"
  "event" : "onConferenceCreated",
  "result" : 1,
  "conferenceOwner" : "ivanov@trueconf.com"
}
```

Parameters description:

- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.

confId – unique conference ID on the server.

- confName – conference name. This parameter exists only when the group conference is started.
- conferenceOwner – This parameter is present only if the terminal is in a group conference and indicates the conference owner.
- confType – indicates a conference type. It can be
  * p2p = 0 (video call)
  * symmetric = 1 (symmetric)
  * asymmetric = 2 (asymmetric)
  * role = 3 (role-based)
- result – a result which can have the following values:

```
NO_ENOUGH_RESOURCES_FOR_CONFERENCE = 0,
CONFERENCE_CREATED_OK = 1,
CREATE_ACCESS_DENIED = 2,
CREATE_HAVE_NO_MONEY = 3,
VSTRCL_CONF_OK = 8320,
VSTRCL_CONF_CALL = 4224,
VSTRCL_CONF_NOTCR = 4225,
VSTRCL_CONF_NOTEX = 8322,
VSTRCL_CONF_ACCDEN = 4227,
VSTRCL_CONF_HNOMON = 4228,
VSTRCL_CONF_PBUSY = 4229,
VSTRCL_CONF_INVP = 4230,
VSTRCL_CONF_PNOTAV = 4231,
VSTRCL_CONF_REJBYPART = 4232,
VSTRCL_CONF_CONFBUSY = 4233,
VSTRCL_CONF_INVCONF = 4234,
VSTRCL_CONF_EXPIRED = 4235,
VSTRCL_CONF_ACCEPTREJ = 4236,
VSTRCL_CONF_NOT_STARTED = 4237,
VSTRCL_CONF_LOCALBUSY = 4238,
VSTRCL_CONF_NAMEUSED = 4239,
```

VSTRCL_CONF_INVMULTI = 4240,
VSTRCL_CONF_INVPARAM = 4241,
VSTRCL_CONF_ONLINELIMIT = 4242,
VSTRCL_CONF_PASSREQ = 4243,
VSTRCL_CONF_PASSWRONG = 4244,
VSTRCL_CONF_NOFRIEND = 4245,
VSTRCL_CONF_BADRATING = 4246,
VSTRCL_CONF_USERTIMEOUT = 4247,
VSTRCL_CONF_IAM_HOST = 4256,
VSTRCL_CONF_SND_FLTR = 8368,
VSTRCL_CONF_INVITEREPLY = 4288

onConferenceDeleted

Description: a notification you receive
* after the video call is finished
* after a group conference is finished

Example:

```
{
  "event" : "onConferenceDeleted",
  "confId" : "00042f28@ru9.trueconf.net#as"
  "result" : 1
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
confId – unique conference ID on the server
- result – a result which can have the following values:
* CONFERENCE_DOESNT_EXIST = 0,
* CONFERENCE_DELETED_OK = 1

onContactBlocked

Description: a notification you receive right after the terminal adds a contact into the blacklist.

Example:

```
{
  "peerId" : "petrov@trueconf.com",
  "event" : "onContactBlocked"
}
```

Parameters description:
- event – field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – unique ID (TrueConfID) of the contact added into the blacklist.

onContactDeleted

Description: a notification received right after your terminal removes a contact from the Address Book.
Example:

```
{
  "peerId" : "petrov@trueconf.com",
  "event" : "onContactDeleted"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – unique ID (TrueConfID) of the contact removed from the Address Book.

onContactUnblocked

Description: a notification you receive right after your terminal removes a contact from the blacklist.
Example:

```
{
  "peerId" : "petrov@trueconf.com",
  "event" : "onContactUnblocked"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – unique ID (TrueConfID) of the contact removed from the blacklist.

onHardwareChanged

Description: a notification you receive after you change your hardware.
Example:

```
{
  "audioCapture" : [
    {
      "description" : "Microphone (B910 HD Webcam)",
      "name" : "Microphone (B910 HD Webcam)"
    },
    {
      "description" : "Microphone (Realtek High Definition Audio)",
      "name" : "Microphone (Realtek High Definition Audio)"
    }
```

```
  ],
  "audioCaptureName" : "Microphone (Realtek High Definition Audio)",
  "audioPlayback" : [
    {
      "description" : "Speakers (Realtek High Definition Audio)",
      "name" : "Speakers (Realtek High Definition Audio)"
    }
  ],
  "audioPlaybackName" : "Speakers (Realtek High Definition Audio)",
  "event" : "onHardwareChanged",
  "videoCapture" : [
    {
      "description" : "Logitech B910 HD Webcam",
      "name" : "Logitech B910 HD Webcam"
    },
    {
      "description" : "USB2.0 UVC HD Webcam",
      "name" : "USB2.0 UVC HD Webcam"
    }
  ],
  "videoCaptureName" : "Logitech B910 HD Webcam"
}
```


Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- audioCapture – the list of available audio capture devices.
- audioPlayback – the list of available audio playback devices.
- videoCapture – the list of available video capture devices.
- name – hardware name which is encoded in BASE64 sequence.
- description – hardware description which is encoded in BASE64 sequence.
- audioPlaybackName – the name of current audio playback device which is encoded in BASE64 sequence.
- audioCaptureName – the name of current audio capture device which is encoded in BASE64 sequence.
- videoCaptureName – the name of current video capture device which is encoded in BASE64 sequence.

onDetailInfo

Description: a notification you receive after user's detailed information has been requested.
Example:


```
{
   "event": "onDetailInfo",
   "peerId": "ivanov@trueconf.com",
   "displayName": "Ivan Ivanov",
   "firstName": "Ivan",
```

```
   "lastName": "Ivanov",
   "mobilePhone": "452452452",
   "workPhone": "45543453",
   "homePhone": "4452872",
   "company": "TrueConf"
}
```


Parameters description:
- peerId – a unique user ID (TrueConfID).
- firstName – first name which is encoded in BASE64 sequence.
- lastName – last name which is encoded in BASE64 sequence.
- displayName – display name which is encoded in BASE64 sequence.
- mobilePhone – mobile phone.
- workPhone – work phone.
- homePhone – home phone.
- company – company's name which is encoded in BASE64 sequence.
- event - a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.

onDeviceModesDone

Description: a notification you receive after pin codes and modes of current video device have been ready.
Example:


```
{
  "videoCaptureName" : "Avermedia C127"
  "activemode" : "NTSC_433",
  "activepin" : "RGB",
  "event" : "commandExecution",
  "modeList" : [
    "NTSC_433",
    "NTSC_M",
    "NTSC_M_J",
    "PAL_60",
    "PAL_B",
    "PAL_D",
    "PAL_H",
    "PAL_I",
    "PAL_M",
    "PAL_N",
    "SECAM_B",
    "SECAM_D",
    "SECAM_G",
    "SECAM_H",
    "SECAM_K",
```

```
    "SECAM_K1",
    "SECAM_L",
    "SECAM_L1"
  ],
  "pinList" : [ "RGB", "Serial Digital" ]
}
```

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- videoCaptureName – video capture device name. You can use it if the device has changed while pin codes have been processed and sent.
- activemode – active mode of the device
- activepin – active pin of the device
- modeList – the list of modes available for device.

- pinList – the list of pin codes available for device.
- supportPTZ – PTZ control support.

onInviteReceived

Description: a notification you receive when you are invited to take part in
    * a group conference
    * a video call

Example:

```
{
  "confPassword" : "",
  "event" : "onInviteReceived",
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com",
  "type" : 0
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – a unique ID TrueConfID of the user who has sent you an invitation.
- peerDn – a display name of the user who has sent you an invitation. It is encoded in BASE64 sequence and needs reverse decoding.
- confPassword – a password required to enter the conference.
- type – conference type which can have the following values:
    * 0 – video call
    * 1 – group conference

onInviteRequestSended

Description: a notification indicating a request to enter the group conference. It is sent by your terminal.
Example:

```
{
  "event" : "onInviteRequestSended",
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com"
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – a unique user ID TrueConfID. This user is the owner of the conference you request to join.
- peerDn – a display name of the user who is the owner of the conference you request to join. It is encoded in BASE64 sequence and needs reverse decoding.

onInviteSended

Description: a notification you receive after
    - your terminal makes a video call to another user.
    - your terminal sends requests to another user to join your group conference.
Example:

```
{
  "event"  : "onInviteSended",
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com",
  "confType"   : 0
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – unique ID TrueConfID, of the user who receives an invitation.
- peerDn – a display name of the user who receives an invitation. It is encoded in BASE64 sequence and needs reverse decoding.
- confType – conference type. It can be
    * 0 – video call
    * 1 – group conference

onLogin

Description: a notification you receive when you has been logged in successfully.
Example:

```
{
  "event" : "onLogin",
  "peerDn" : "BG DN",
  "peerId" : "ivanov3@trueconf.com",
  "result" : 0,
}

{
  "event" : "onLogin",
  "result" : 3
}
```

Parameter description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- peerId – your unique user ID TrueConfID
- peerDn – Your display user name. It is encoded in BASE64 sequence and needs reverse decoding.
- result – authorization result. It can be:
  * USER_LOGGEDIN_OK = 0, (login successful, otherwise error code)
  * USER_ALREADY_LOGGEDIN = 1, (answer on CheckUserLoginStatus_Method, if current CID is already authorized at TransportRouter)
  * NO_USER_LOGGEDIN = 2, (answer on CheckUserLoginStatus_Method, if current CID is not authorized at TransportRouter - can try to login)
  * ACCESS_DENIED = 3, (incorrect password or other problems with DB)
  * SILENT_REJECT_LOGIN = 4, (client shouldn't show error to user (example: incorrect AutoLoginKey))
  * LICENSE_USER_LIMIT = 5, (license restriction of online users reached, server cannot login you)
  * USER_DISABLED = 6, (user exist, but he is disabled to use this server)
  * RETRY_LOGIN = 7, (client should retry login after timeout (value in container or default), due to server busy or other server problems)
  * INVALID_CLIENT_TYPE = 8 (user cannot login using this client app (should use other type of client app))

onLogout

Description: a notification you receive after logging out.
Example:

```
{
  "cause" : 0,
```

```
  "event" : "onLogout",
  "result" : 0
}
```

Parameters description:
- event – a field indicating message type (event or command). It could be either commandExecution or onNotification (update notification),e.g. onInviteSended.
- result – a result of logging out. It can have the following values:
  * USER_LOGGEDOUT_OK = 0,
  * USER_ALREADY_LOGGEDOUT = 1
- cause – the cause of logging out. It can have the following values:
  * USER_LOGGEDOUT_BY_REQUEST = 0,
  * USER_LOGGEDIN = 1

onRecordRequest

Description: a notification you receive when the user requires your permission to record your video stream

Example:

```
{
  "event" : "onRecordRequest",
  "peerId" : "ivanov@trueconf.com"
  "peerDn" : "Ivan Ivanov"
}
```

Parameters description:

- onRecordRequest – a field indicating the type of message (event or command).
It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – TrueConfID unique identifier of the user who requires permission to record your video stream.
- peerDn - user display name. It is encoded in BASE64 sequence and needs reverse decoding.

onRecordRequestReply

Description: a notification you receive when you accept recording of your video stream.

Example:

```
{
  "event" : "onRecordRequestReply",
```

```
  "peerId" : "ivanov@trueconf.com",
  "recordAllowed" : true
}
```

Parameters description:

- onRecordRequest - a field indicating the type of message (event or command).
  It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId - TrueConfID unique identifier of the user who requires permission to record your video stream.
- recordAllowed – a flag indicating permission to record.

onRejectReceived

Description: a notification you receive when the user rejects your invitation to video call or after the terminal you sends request to participate in a conference and is rejected.

Example:

```
{
  "cause" : 0,
  "event" : "onRejectReceived",
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com"
}
```

Parameters description:

- event – a field indicating the type of message (event or command).
  It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – TrueConfID unique identifier of the user who rejects your call.
- peerDn – display name of the user who rejects your call. It is encoded in BASE64 sequence and needs reverse decoding.
- cause – a cause for rejection which can have the following values:

```
  * REJECTED_BY_PARTICIPANT       = 0,
  * CONFERENCE_IS_BUSY            = 1,
  * PARTISIPANT_IS_BUSY           = 2,
  * PARTISIPANT_NOT_AVAILABLE_NOW  = 3,
  * INVALID_CONFERENCE            = 4,
  * INVALID_PARTISIPANT           = 5,
  * JOIN_OK                      = 6,
  * REACH_MONEY_LIMIT            = 7,
  * REJECTED_BY_ACCESS_DENIED     = 8,
  * REJECTED_BY_LOGOUT           = 9,
  * REJECTED_BY_RESOURCE_LIMIT    = 10,
  * REJECTED_BY_LOCAL_RES        = 11,
  * CONFERENCE_PASSWORD_REQUIRED  = 12,
  * REJECTED_BY_WRONG_PASSWORD    = 13,
```

* REJECTED_BY_NOFRIEND = 14,
* REJECTED_BY_BADRATING = 15,
* REJECTED_BY_TIMEOUT = 16,
* THIS_IS_CONFERENCE = 17

onRejectSent

Description: a notification you receive when
*the terminal rejects incoming video call
*the terminal rejects invitation to participate in your conference
*the terminal rejects an incoming request to participate in someone else's conference

Example:

```
{
  "cause" : 0,
  "event" : "onRejectSent",
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com"
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.

- peerId – TrueConfID unique identifier of the user whose call you reject.
- peerDn – display name of the user whose call you reject. It is encoded in BASE64 sequence and needs reverse decoding.
- cause – a cause for rejection, which can have the following values:
  * REJECTED_BY_PARTICIPANT = 0,
  * CONFERENCE_IS_BUSY = 1,
  * PARTISIPANT_IS_BUSY = 2,
  * PARTISIPANT_NOT_AVAILABLE_NOW = 3,
  * INVALID_CONFERENCE = 4,
  * INVALID_PARTISIPANT = 5,
  * JOIN_OK = 6,
  * REACH_MONEY_LIMIT = 7,
  * REJECTED_BY_ACCESS_DENIED = 8,
  * REJECTED_BY_LOGOUT = 9,
  * REJECTED_BY_RESOURCE_LIMIT = 10,
  * REJECTED_BY_LOCAL_RES = 11,
  * CONFERENCE_PASSWORD_REQUIRED = 12,
  * REJECTED_BY_WRONG_PASSWORD = 13,
  * REJECTED_BY_NOFRIEND = 14,
  * REJECTED_BY_BADRATING = 15,
  * REJECTED_BY_TIMEOUT = 16,
  * THIS_IS_CONFERENCE = 17

onRemarkCountDown

Description: a notification that informs you about remaining time for making a remark in a role-based conference.
Example:

```
{
  "event" : "onRemarkCountDown",
  "peerId" : "ivanov3@trueconf.com",
  "timeout" : 2
}
```

Parameters description:
- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – TrueConfID unique identifier of the user who accepts the call.
- timeout – remaining time for making a remark in seconds.

onRequestInviteReceived

Description: a notification which informs you about a request from another user to join your existing group conference.
Example:

```
{
  "event" : "onRequestInviteReceived",
  "peerDn" : "Ivan Ivanov",
  "peerId" : "ivanov@trueconf.com"
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended
- peerId – TrueConfID unique identifier of the user who wants to join your conference.

- peerDn – display name of the user who wants to join your conference. It is encoded in BASE64 sequence and needs reverse decoding.

onRoleEventOccured

Description: a notification you receive after one of the conference participants changes his or her role: takes or leaves the podium, etc.
Example:

```
{
  "broadcast" : false,
```

```
  "event" : onRoleEventOccured,
  "peerId" : "ivanov@trueconf.com",
  "peerDn" : "Ivan Ivanov",
  "result" : 1,
  "role" : 2,
  "slideShowFlag" : false,
  "type" : 3,
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – TrueConfID unique identifier of the user who accepts the call.
- peerDn – user display name. It is encoded in BASE64 sequence and needs reverse decoding.

- role - participant's role:
  * PR_EQUAL = 0,
  * PR_COMMON = 1,
  * PR_LEADER = 2,
  * PR_PODIUM = 3,
  * PR_REPORTER = 4,
  * PR_REMARK = 5

- type - type of role change:
* RET_INQUIRY = 0,
* RET_ANSWER = 1,
* RET_CONFIRM = 2,
* RET_NOTIFY = 3

- result – a result you receive after the request to change a role:
* RIA_POSITIVE = 0,
  * RIA_COMMON = 1,
  * RIA_ROLE_BUSY = 2,
  * RIA_BY_PARTICIPANT = 3,
  * RIA_PATRICIPANT_BUSY = 4,
  * RIA_PARTICIPANT_ABSENT = 5

- broadcast – a flag indicating if the conference participant is taking the podium right now.
- slideShowFlag – a flag indicating if the conference participant has started the slide show. It can be true or false.

onSelfSSInfoUpdate

Description: a notification you receive after slide show information has been updated. This update applies **only** to the terminal managed by you. You will receive this notification when there is a possibility to update slideshow features or change the list of slideshow images.
Example:

```
{
  "canUse" : false,
  "event" : "onSelfSSInfoUpdate",
}

{
  "canUse" : true,
  "event" : "onSelfSSInfoUpdate",
  "imgs" : [
    {
      "path" : "/home/ivanov/images/1.jpg"
    },
    {
      "path" : "/home/ivanov/images/stampPNG.png"
    }
  ]
}
```

Parameters description:

- event - a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- canUse - a flag indicating possibility to use slideshow.
* true – available.
* false - not available.
- imgs - a flag indicating the list of slideshow images. The flag is installed in your terminal via GUI.
  path - image path.

onServerConnected

Description: a notification you receive after you have successfully connected to a server or a service.
Example:

```
{
  "event" : "onServerConnected",
  "server" : "ru9.trueconf.net",
  "port" : 80,
  "service" : true
}
```

Parameters description:
- event – a field indicating the type of message (event or command).

It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- server – the address of the server which is connected to your terminal. If the client is connected to the server, it is ip address, if the client is connected to the service, it is service name.
- port – a server port to which the terminal is connected.
- service – a flag indicating a type of connection.
* true – the terminal is connected to Online service.
* false – the terminal is connected to the Server.

onServerDisconnected

Description: a notification you receive after the terminal has been disconnected from TrueConf Online service or from the server. You will receive a notification after a while, because the application will try to connect to TrueConf Online service or to the server for some time.
Example:

```
{
  "event" : "onServerDisconnected",
  "server" : "ru9.trueconf.net",
  "service" : true
  "port" : 4307
}
```

Parameters description:
- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- server – the address of the server to which the terminal is connected to.
- port – the port of the server to which the terminal is connected to.
- service – a flag indicating whether the client is connected to the server or to the service.

onSettingsChanged

Description: a notification you receive after the settings of your terminal have been changed
Example:

```
{
   "event" : "onSettingsChanged",
   "settingName" : value
}
```

Parameters description:
- event – a field indicating the type of message (event or command).

It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- value – new setting value. It can be a string parameter, a number, etc., depending on the setting type.
- settingName – one of the application settings. It can be:
- addressBookOnlaineFilter – display only those users who are online now. Itcan be true, false.
- appRunAtStartupKey – run the application at the system startup, it can be true, false.
- audioPlayLevel – audio playback level The range of values is 0.00 - 1.0.
- audioRecordLevel – audio capture level. The range of values is 0.00 - 1.0.
- sndAppStart – start audio playback at the system startup. It can be true, false.
- appShowUserNameLabels – display the names of conference participants. It can be true, false.
- appShowUserNotifications – display participants notifications. It can be true, false
- autoAccept - automatically accept incoming calls. It can be true, false.
- autoEnterConference – allow access to the conference to everyone. It can be true, false.
- selfViewMirror – mirrored self-view It can be true, false.
- enableAutologin – enable autologin. It can be true, false.
- inputBandWidth – input bandwidth. It can be 32, 64, 128, 256, 512, 1024, 2048.

- language - localization. It can be:
* 1 - Russian locale,
* 2 - others

- lastSelectedConference - the last selected conference type. It can be:
* symmetric = 0
(symmetric)
* asymmetric = 1
(asymmetric)
* role = 2
(role-based)
- leaveAppRunningInTray – keep the application running when you close it and minimize it to the tray. It can be true, false.
- outputBandWidth - output bandwidth. It can be 32, 64, 128, 256, 512, 1024, 2048.
- rejectCallsNotFromAB – reject calls from the users who are not included into the Address Book. It can be true, false.
- sndBusy – play back busy Call sound. It can be true, false.
- sndCallRing – Play back call ring sound. It can be true, false.
- sndCallStart – Play back call start sound. It can be true, false.
- sndInChat – play back incoming message sound in a chat. It can be true, false.

- sndOutChat – play back outgoing message sound in a chat. Itcan be true, false.
- sndServerDisconnect – play back server disconnection sound. It can be true, false.
- videoFltrFreq – network frequency.

onSlideShowStart

Description: slideshow notification.
Example:

```
{
  "event" : "onSlideShowStart",
  "peerId" : "petrov@trueconf.com",
  "peerDn" : "Peter Petrov",
  "title" : "Slideshow11 1/1"
}
```

Parameters description:
- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – TrueConfID unique user identifier
- peerDn – user display name. It is encoded in BASE64 sequence and needs reverse decoding
- title – for example, "Slideshow11 1/1" indicates **Slideshow11** slideshow and **1/1,** the first slide out of one. It is encoded in BASE64 sequence.

onSlideShowStop

Description: a notification you receive after the slideshow has been finished by one of the participants.

Example:

```
{
  "event" : "onSlideShowStop",
  "peerId" : "petrov@trueconf.com"
}
```

Parameters description:
- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – TrueConfID unique identifier of the user who finished slideshow.

onSSInfoUpdate

Description: a notification you receive when you change slideshow parameters.
Example:

```
{
  "canUse" : true,
  "event" : "onSSInfoUpdate",
  "currentSlide" : "1.jpg",
  "imgs" : [
{
    "img" : "/home/ivanov/images/1.jpg"
  },
  {
    "img" : "/home/ivanov/images/1 (another copy).jpg"
  },
  {
    "img" : "/home/ivanov/images/1 (copy).jpg"
  }
],
    "running" : false
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- canUse – a flag indicating if you can start slideshow. It can be true or false.
  - currentSlide – a current slide. It can be absent if none of the slides has been selected.
  - imgs – the list of slides for slideshow. The order of this list and the order stated in the client are identical. A flag is present only when the list is not empty.
  - img – image location.
- running – a flag indicating slideshow state. It can be true (slideshow is running) or false (slideshow is not running).

onStopCalling

Description: a notification you receive when a p2p call is finished:
*For example, if you decide to cancel the call and click the Cancel button during the call.
*A similar situation, but in this case the user calling you decides to cancel the call.

Example:

```
{
  "event" : "onStopCalling",
  "peerId" : "petrov@trueconf.com",
  "peerDn" : "Ivan Ivanov"
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – unique user identifier (TrueConfID)
- peerDn – user display name. It is encoded in BASE64 sequence and needs reverse decoding.

onUidTtlAboutToFinished

Description: a notification you receive one minute before your uid is deleted due to inactivity. To extend it, you need to run extendUidTtl command.
Example:

```
{
   "event": "onUidTtlAboutToFinished",
   "cid": "sIcY0Jmn4xy3c9sZ"
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended
- cid – unique connection ID.

onUpdateAvatar

Description: a notification you receive when user avatar has been updated (the latest relevant avatar is received).
Example:

```
{
   "event": "onUpdateAvatar",
   "peerId": "ivanov@trueconf.com",
   "avatar": "base64 avatar src"
}
```

Parameters description:
- peerId – unique user identifier (TrueConfID)
- avatar – image source.
- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.

onUpdateCameraInfo

Description: a notification you receive when the format and resolution of the captured video has been changed.
Example:

```
{
   "event": "onUpdateCameraInfo",
  "width": 1280,
   "height": 720,
   "fps": 1500,
   "format" : 1
}
```

Parameters description:
- width – width of the captured video.
- height – height of the captured video.
- fps – the number of frames per second.
- format – video format. It can be:
  * YUYV = 0,
  * YUY2 = 1,
  * YVYU = 2,
  * MJPG = 3,
  * I420 = 4,
  * IYUV = 5,
  * UYVY = 6,
  * HDYC = 7,
  * YV12 = 8,
  * NV12 = 9,
  * NV16 = 10,
  * NV21 = 11,
  * RGB32 = 12,
  * RGB24 = 13,
  * ARGB = 14,
  * BGRA = 15,
  * YUV444 = 16,
  * H264 = 17,
  * H264_ES = 18,
  * H265 = 19,
  * VP80 = 20,
  * VP90 = 21,
  * STR0 = 22,
  * I420_STR0 = 23,

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.

onUpdateConferenceParticipants

Description: a notification you receive after a new participant joined the conference, left the conference, etc.
Example:

```
{
  "Participants" : [
    {
      "peerDn" : "Peter Petrov",
      "peerId" : "petrov@trueconf.com",
      "mic" : true,
      "video" : false
    },
    {
      "peerDn" : "Ivan Ivanov",
      "peerId" : "ivanov@trueconf.com",
      "mic" : false,
      "video" : false
    }
  ],
  "event" : "onUpdateConferenceParticipants"
}
```

For group conference:

```
{
  "Participants" : [
    {
      "broadcast" : true,
      "peerDn" : "Petr Petrov",
      "peerId" : "petrov@trueconf.com",
      "role" : 1,
      "slideShowFlag" : false,
      "mic" : false,
      "video" : false
    },
    {
      "broadcast" : true,
      "peerDn" : "Ivan Ivanov",
      "peerId" : "ivanov@trueconf.com",
      "role" : 2,
      "slideShowFlag" : false,
      "mic" : true,
      "video" : true
    }
  ],
  "event" : "onUpdateConferenceParticipants"
}
```

Parameters description:

- event – a field indicating the type of message (event or command).

It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – unique user identifier (TrueConfID).
- role – participant's role:
 * PR_EQUAL = 0,
 * PR_COMMON = 1,
 * PR_LEADER = 2,
 * PR_PODIUM = 3,
 * PR_REPORTER = 4,
 * PR_REMARK = 5

- broadcast – a flag indicating if the conference participant is taking the podium right now.
- slideShowFlag – a flag indicating if the conference participant has started the slide show. It can be true or false
- mic – microphone state. It can be switched on (true) or switched off (false).
- video – camera state. It can be switched on (true) or switched off (false).

onTariffRestrictionsChanged

Description: a notification you receive after the application permission type has been changed: conference initiation, maximum number of participants in a specific conference, etc.
Example:


```
{
  "asymMaxNumber" : 0,
  "call" : 1,
  "commonMulti" : 1,
  "createMulti" : 1,
  "event" : "onTariffRestrictionsChanged",
  "needPassword" : 1,
  "rlMaxNumber" : 0,
  "roleMaxNumber" : 0,
  "symMaxNumber" : 3,
  "canUseSlideShow": 1,
  "canUseDesktopSharing": 1,
  "canChangeAddressBook": 1,
  "canEditGroups": 1,
  "canUseDialer": 0,
}
{
  "asymMaxNumber" : 16,
  "call" : 1,
  "commonMulti" : 1,
  "createMulti" : 1,
  "event" : "onTariffRestrictionsChanged",
  "needPassword" : 1,
```

```
  "rlMaxNumber" : 4,
  "roleMaxNumber" : 120,
  "symMaxNumber" : 9,
  "canUseSlideShow": 1,
  "canUseDesktopSharing": 1,
  "canChangeAddressBook": 1,
  "canEditGroups": 1,
  "canUseDialer": 1,
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- asymMaxNumber – maximum number of participants in an asymmetric conference (if 0, you are not allowed to start this type of conference).
- call – a parameter indicating if the user can make a video call.
- commonMulti – a parameter indicating if the usercan start a multipoint conference.
- createMulti – a parameter indicating if the user is allowed to start a multipoint conference.
- needPassword – a parameter indicating if the password is required in a multipoint conference
- rlMaxNumber – the maximum number of conference participants that can take the podium at the same time (if 0, you are not allowed to start this type of conference).
- roleMaxNumber – the maximum number of conference participants in a role-based conference (if 0, you are not allowed to start this type of conference).
- symMaxNumber – the maximum number of participants in a symmetric conference (if 0, you are not allowed to start this type of conference).
- canUseSlideShow – a parameter indicating if the slide show is available.
- canUseDesktopSharing – a parameter indicating if desktop sharing is available.
- canChangeAddressBook – a parameter indicating if the Address Book can be changed.
- canEditGroups – a parameter indicating if the user groups can be edited.
- canUseDialer – a parameter indicating if the dialer is available.

onVideoMatrixChanged

Description: a notification you receive when the matrix type and,optionally, video layouts and slots are changed.

Example:

```
  "event" : "onVideoMatrixChanged",
  "mainWindowHeight" : 544,
  "mainWindowWidth" : 960,
  "matrixType" : 4,
```

```
  "participants" : [
    {
      "height" : 181,
      "left" : 0,
      "peerId" : "test2@ub0n3.trueconf.name",
      "priority" : 0,
      "slotId" : 0,
      "top" : 363,
      "width" : 319
    },
    {
      "height" : 544,
      "left" : 0,
      "peerId" : "test1@ub0n3.trueconf.name",
      "priority" : 1,
      "slotId" : 1,
      "top" : 0,
      "width" : 960
    }
  ]
}
```

Parameters description:

- event – a field indicating the type of message (event or command). It can be either commandExecution (command execution) or onNotification (update notification), for example onInviteSended.
- peerId – unique user identifier (TrueConfID)
- participants – the list of slots and conference participants.
- slotId – the number of slot.
- top – top coordinates.
- width – window width.
- height – window height.
- mainWindowWidth – main window width.
- mainWindowHeight – main window height.
- left - left-handed coordinates.
- priority – priority. It can be:
* PRIORITY_LOW = 0
* PRIORITY_HIGH = 1

- matrixType – matrix type. It may have the following types:
* even= 0, all windows are of the same size (for multipoint conference)
* big = 1, one window is big, while the others are small (for multipoint conference)
* one = 2, display only the video of the first user in the participants list (for any type of conference)
* p2p = 3, display desktop sharing, slideshow, participant's video and self-view (for p2p calls)
* oneSelf= 4, display participant's big video and small self-view in the corner (for p2p calls)

* PSshow = 5, display slideshow, participant's video and a self-view (for p2p calls)

onJabraHookOffPressed

Description: a notification you receive after you press the "Call" button on Jabra speakerphone. The parameters are absent.

onJabraHangUpPressed

Description: a notification you receive after you press the "Hang Up" button on Jabra speakerphone. The parameters are absent.

Receive slides and conference frames

To receive an avatar, you need to send Get request.
Request URL: url/type/peerId
Request URL sample: http://localhost:1234/conf/ivanov@qa.trueconf.net
Input parameters description:
- url – the address which is used by the terminal to send images.
- type – a type which indicates if the avatar (avatar) or the frame (conf) have been received.
- peerId – unique user ID (TrueConfID).

OnSlideShowInfoUpdate

Description: The notification that comes when one of the slideshow information settings is changed.
Example:

```
{
 "event" : "onSlideShowInfoUpdate"6
  "available": true,
  "started": false,
  "slides": [
     {
        "idx": 0,
        "name": "U2xpZGUgMDAxLmpwZw=="
     },
     {
        "idx": 1,
        "name": "U2xpZGUgMDAyLmpwZw=="
     },
     {
        "idx": 2,
        "name": "MS5qcGc="
     },
```

```
      {
         "idx": 3,
         "name": "MS5wbmc=",
         "url" : "url in base64"
      }
   ],
   "currentSlideIdx": 0,
   "title": "U2FtcGxlUFBUWC5wcHR4",
}
```

Parameters description:
- event – a field which indicates message type (event or command). It can be either: commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- available – a flag which indicates that the slideshow can be started.
- started – a flag which indicates if the slideshow is running.
- currentSlideIdx – present only if the slideshow is running and indicates the index of the slide currently displayed.
- title – present only if the slideshow is running and indicates the slideshow title. It is Base46-encoded and needs reverse decoding.
- idx – present only if the slideshow is running and indicates the slide sequence number.
- name – present only if the slideshow is running and indicates the slide filename. It is BASE64-encoded and needs reverse decoding.
- url – present only if the slideshow is running and indicates the downloaded slide link. It is BASE64-encoded and needs reverse decoding.

onBroadCastPictureStateChanged

Description: The notification that comes when the picture broadcasted instead of video changes its state:

```
{
   "event": "onBroadcastPictureStateChanged",
   "filename": "broadcast_img"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), for example onInviteSended.
- filename – filename (without the path to it) which contains the picture used for broadcasting. Empty line is displayed when the picture broadcasted instead of video is disabled.

onMonitorsInfoUpdated

Description: The notification that comes when changing information about monitors: list, current index, etc.:

```
{
   "event": "onMonitorsInfoUpdated",
   "monitors": [
        {
           "name":
"R2VuZXJpYyBQblAgTW9uaXRvciAoTlZJRElBIEdlRm9yY2UgMjEwICk=",
           "primary": true,
          "index": 0,
           "geomMonitor": {
              "x": 0,
              "y": 0,
              "width": 1920,
              "height": 1080
           },
           "geomWork": {
              "x": 0,
              "y": 0,
              "width": 1920,
              "height": 1040
           }
        },
        {
           "name":
"R2VuZXJpYyBQblAgTW9uaXRvciAoSW50ZWwoUikgSEQgR3JhcGhpY3MgNDYwMCk=",
           "primary": false,
           "index": 1,
           "geomMonitor": {
              "x": -1024,
              "y": 0,
              "width": 1024,
              "height": 768
           },
           "geomWork": {
              "x": -1024,
              "y": 0,
              "width": 1024,
              "height": 728
           }
        }
     ],
   "currentMontior": 1,
}
```


Parameters description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g.onInviteSended.

- name – the name of the monitor (with the driver used in brackets). It is sent in BASE64 sequence and needs reverse decoding.
- primary – a flag that indicates if it is primary monitor.
- x – x-coordinate.
- y – y-coordinate.
-- width – width.
- height – height.
- index – monitor index.
-currentMonitor – index of the monitor where the terminal is located.

onCallHistoryUpdated

Description:
The notification that comes when the call history is changed: missed calls, held calls, etc. Please note that the list includes only those calls which have been made after the date stated in the lastView parameter.

```
{
  "event": "onCallHistoryUpdated",
  "calls": [
     {
        "type": 2,
        "confType": 0,
        "peerId": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
        "peerDn": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
        "startTime": 1493379174,
        "duration": 1,
        "viewed": false
     },
     {
        "type": 2,
        "confType": 0,
        "peerId": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
        "peerDn": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
        "startTime": 1493379169,
        "duration": 1,
        "viewed": false
     }
   ],
   "lastView": 1493379036
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), for example onInviteSendedю
- lastView – the date up to which the calls were viewed i.e. calls and conferences held after this date will be shown in the list, the rest will not be displayed.
- confType – conference type. There are two types:

  * p2p = 0,
  * group = 1
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.
- peerDn – display user name. It is sent in BASE64 sequence and needs reverse decoding.
- startTime – starting time
- duration duration
- viewed – flag which indicates if the call has been "viewed".
- type – type of call. There are two types:

  * incoming = 0,
  * outgoing = 1,
  * notReceived = 2

onAudioCapturerRmsLevelUpdated

Description: The notification that comes during the microphone test.

```
{
   "event": "onAudioCapturerRmsLevelUpdated",
   "lvl": 0.5850950479507446
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), for example onInviteSended.
- lvl – volume of the captured sound.

onCommandReceived

Description: The notification that comes after the terminal has received a remote command.

```
{
   "event": "onCommandReceived",
   "peerId": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
   "command": "remote:mic_off"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.
- command – command.

onCommandSent

Description: The notification that comes after the terminal you manage sent a command to a remote client:

```
{
  "event": "onCommandSent",
  "peerId": "ZmV0dGVybGVzczJAdHJ1ZWNvbmYuY29t",
  "command": "remote:mic_off"
}
```

Parameters Description:

- event – a field which indicates message tyoe (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.
- command – command.

onToneDial

Description: Notification that comes when the symbol has been sent (extension dialing).

```
{
   "event": "",
   "symbol": "5"
}
```

Parameters Description:

- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.
- symbol – sent symbol.

onCmdAddToAbook

Description: notification about the outgoing command to add a user to the address book.

```
{
   "event": "onCmdAddToAbook",
   "peerId": "ZmV0dGVybGVzczNAdHJ1ZWNvbmYuY29t",
   "peerDn": "Qm9nZGFu"
}
```

Parameters Description:

- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.
- peerDn – display user name. It is sent in BASE64 sequence and needs reverse decoding.

onCmdRenameInAbook

Description: notification about the outgoing command to rename the user in the address book.

```
{
   "event": "onCmdRenameInAbook",
   "peerId": "eml6YUB0cnVlY29uZi5jb20=",
   "peerDn": "Wml6YS3Qt9C40LfQsA=="
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onCmdRemoveFromAbook

Description: Notification about outgoing command to delete the user from the address book.

```
{
   "event": "onCmdRemoveFromAbook",
   "peerId": "ZmV0dGVybGVzczNAdHJ1ZWNvbmYuY29t"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). There may be either a commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onCmdBlock

Description: notification about the outgoing command to add a user to the black list.

```
{
   "event": "onCmdBlock",
   "peerId": "eml6YUB0cnVlY29uZi5jb20="
```

```
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onCmdUnBlock

Description: notification about the outgoing command to delete a user from the black list.

```
{
   "event": "onCmdUnblock",
   "peerId": "eml6YUB0cnVlY29uZi5jb20="
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either a commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onCmdCreateGroup

Description: Notification about the outgoing command to create a group.

```
{
   "event": "onCmdCreateGroup",
   "name": "base64EncodedName"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either a commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- name –- the name of the group. It is sent in BASE64 sequence and needs reverse decoding.

onCmdCreateGroup

Description: Notification about an outgoing command to delete a group.

```
{
   "event": "onCmdRemoveGroup",
   "id": 3159
}
```

Parameters Description:

- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – group id.

onCmdRenameGroup

Description: Notification about the outgoing command to rename a group.

```
{
  "event": "onCmdRenameGroup",
  "id": 31596
  "name" : "base64EncodedName"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – group id.

onCmdAddToGroup

Description: notification about the outgoing command to add a user to the group.

```
{
  "event": "onCmdAddToGroup",
  "id": 31596
  "peerId" : "base64EncodedPeerId"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – group id.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onCmdRemoveFromGroup

Description: Notification about the outgoing command to delete a user from a group.

```
{
  "event": "onCmdRemoveFromGroup",
  "id": 31596
  "peerId" : "base64EncodedPeerId"
}
```

Parameters Description:

- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – group id.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onGroupsUpdate

Description: The notification that comes after user group has been changed: member added or removed, group name changed, etc.

```
{
  "event": "onGroupsUpdate",
  "groups": [
     {
        "id": 3163,
        "name": "TEFMQUxB",
        "peers": [
           {
              "peerId": "eml6YUB0cnVlY29uZi5jb20=",
              "peerDn": "Wml6YS3Qt9C40LfQsA=="
           }
        ]
     }
  ]
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – group id.
- name – the name of the group. It is sent in BASE64 sequence and needs reverse decoding.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onChatMessageSent

Description: notification which comes when a personal message to another user has been sent (chat).

```
{
  "event": "onChatMessageSent",
  "peerId": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
  "message": "aGVsbG8sIHdvcmxkIDop"
}
```

Parameters Description:

- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.

onGroupChatMessageSent

Description: Notification which comes when a message is sent during a group conference (group conference chat).

```
{
   "event": "onGroupChatMessageSent",
   "message": "aGVsbG8sIHdvcmxkIDop"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). it can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- message – message. It is sent in BASE64 sequence and needs reverse decoding.

onIncomingGroupChatMessage

Description: Notification which comes when the message has been received in a group conference.

```
{
   "event": "onIncomingGroupChatMessage",
   "peerId": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
   "peerDn": "Qm9nZGFuIEdsaW5za2l5",
   "message": "OkQ=",
   "time": 1493731583
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- message – message. It is sent in BASE64 sequence and needs reverse decoding.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.
- peerDn – display user name. It is sent in BASE64 sequence and needs reverse decoding.
- time – time.

onIncomingChatMessage

Description: The notification which indicates that the personal message has been received.

```
{
   "event": "onIncomingChatMessage",
   "peerId": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20=",
   "peerDn": "Qm9nZGFuIEdsaW5za2l5",
   "message": "TE9M",
   "time": 1493731621
}
```

Parameters Description:
- event – a field which indicates message type (event or command). There may be either a commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- message – message. It is sent in BASE64 sequence and needs reverse decoding.
- peerId – user ID. It is sent in BASE64 sequence and needs reverse decoding.
- peerDn – display user name. It is sent in BASE64 sequence and needs reverse decoding.
- time – time.

onCmdChatClear

Description: Notification which indicates that the message history was cleared out.

```
{
   "event": "onCmdChatClear",
   "id": "ZmV0dGVybGVzc0B0cnVlY29uZi5jb20="
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id - chat id. It is sent in BASE64 sequence and needs reverse decoding.

onReceivedFileRequest

Description: a notification which indicates that the request to receive a file has been sent (file is being sent to you).

```
{
  "event": "onReceivedFileRequest",
  "id": 1,
  "peerId": "Ym9nZGFuMkBzaHR5a2guc2VydmVy",
   "peerDn": "Ym9nZGFuMg==",
   "fileName": "cGl4aWUuZXhl"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – request id.
- peerId – the id of the user who sent you the file. It is sent in BASE64 sequence and needs reverse decoding.
- peerDn – display name of the user who sent you the file. It is sent in BASE64 sequence and needs reverse decoding.
- fileName – the name of the file. It is sent in BASE64 sequence and needs reverse decoding

onFileTransferAvailable

Description: Notification which indicates that the files are available to operate with (sending and receiving).

```
{
   "event": "onFileTransferAvailable",
   "available": false
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either a commandExecution (command execution), or onNotification (update notification), for example onInviteSended.
- available – parameter which indicates that the files are available for transfer. It can be either true or false.

onFileAccepted

Description: Notification which indicates that the command to accept file reception is executed.

```
{
   "event": "onFileAccepted",
   "id": 1
}
```

Parameters Description:
- event – a field which indicates the type of message (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – request id.

onFileRejected

Description: A notification which indicates that the command to refuse to receive a file is executed.

```
{
   "event": "onFileRejected",
```

```
   "id": 1
}
```

Parameters Description:
- event – a field which indicates the type of message (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – request identifier.

onFileStatus

Description: a notification which indicates file task progress (reception / transferring progress):

```
{
   "event": "onFileStatus",
   "id": 1,
   "status": 2
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – request identifier.
- url – a link to the file transferred (previously uploaded to the server using the post method). It is sent in BASE64 sequence and needs reverse decoding.
- status – status. It can be of six types:
  * NotImplemented = 0,
  * StartError = 1,
  * Started = 2,
  * Complete = 3, (receiving / sending file completed)
  * UndefinedError = 4,
  * Pending = 5 (pending agreement / rejection, only available for incoming files).

onFileSent

Description: a notification which indicates that the command to send a file to the user is executed.

```
{
   "event": "onFileSent",
   "name": "cGl4aWUuZXhl",
   "peerId": "Ym9nZGFuMkBzaHR5a2guc2VydmVy",
    "peerDn": "Ym9nZGFuMkBzaHR5a2guc2VydmVy",
   "url":
"aHR0cDovLzE5Mi4xNjguMC4xMDI6MjY2MzYvZmlsZXRyYW5zZmVyL3BpeGllL
mV4ZQ==",
   "id" : '2'
```

```
}
```

Parameters description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- id – request id.
- peerId – id of the user who received the file. It is sent in BASE64 sequence and needs reverse decoding.
- peerDn – display name of the user receiving the file. It is sent in BASE64 sequence and needs reverse decoding.
- name – the name of the file sent. It is sent in BASE64 sequence and needs reverse decoding.
- url – a link to a file pre-uploaded to the server using post method. It is sent in BASE64 sequence and needs reverse decoding.

onFileConferenceSent

Description: a notification which indicates that the command to send a file to all users in the current conference is executed.

```
{
  "event": "onFileSent",
  "name": "cGl4aWUuZXhl",
  "peerId": "Ym9nZGFuMkBzaHR5a2guc2VydmVy",
  "peerDn": "Ym9nZGFuMkBzaHR5a2guc2VydmVy",
  "url":
"aHR0cDovLzE5Mi4xNjguMC4xMDI6MjY2MzYvZmlsZXRyYW5zZmVyL3BpeGllL
mV4ZQ==",
  "id" : '2'
}
```

Parameters Description:
- event - a field which indicates the message type (event or command). It can be either a commandExecution (command execution), or onNotification (update notification), for example onInviteSended.
- id – request id.
- peerId – id of the user receiving the file. It is sent in BASE64 sequence and needs reverse decoding.
- peerDn – display name of the user receiving the file. It is sent in BASE64 sequence and needs reverse decoding.
- name – the name of the file sent. It is sent in BASE64 sequence and needs reverse decoding.
- url – a link to the file pre-uploaded to the server. It is sent in BASE64 sequence and needs reverse decoding.

onDataDeleted

Description: Notification which indicates that the user data container has been removed.

```
{
   "event": "onDataDeleted",
   "containerName": "base64EncodedContainerName"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either a commandExecution (command execution), or onNotification (update notification), e.g. eonInviteSended.
- containerName – the name of the container removed. It is sent in BASE64.

onDataSaved

Description: A notification which indicates that the container with user data has been saved.

```
{
  "event": "onDataDeleted",
  "containerName": "base64EncodedContainerName",
  "data": "base64EncodedData",
   "flag": "def"
}
```

Parameters Description:
- event – a field which indicates message type (event or command). It can be either commandExecution (command execution), or onNotification (update notification), e.g. onInviteSended.
- containerName – the name of the container where the data is stored. It is sent in BASE64.
- data – saved data. It is sent in BASE64.
- flag – the flag under which it was saved. There are 3 types:
    * "def" - if a container with this name already exists, all data will be overwritten (loss of previous data).
    * "beg" - if a container with this name already exists, new data will be written in the beginning (the old data will be saved).
    * "ate" - if a container with this name already exists, new data will be written in the end (the old data will be saved).