Four types of REST methods

There are four types of REST methods

- connect
- direct invocatons
- event
- incoming invocations

The `connect` method determines behavior of all other methods.

The `connect` method should return in the response only those data that it really received from WCS with an exception of the case when some field in the response should be explicitly overwritten or another status must be returned.

Other methods are recommended to return in the response only those data that they received with an exception of the case when some field in the response should be explicitly overwritten or another status must be returned.

All methods should return the `200 OK` HTTP status with an exception of the case when another status should be explicitly returned, for example, `403 Forbidden`.

Direct invocations from WCS JavaScript API

These events may be interrupted by backend server:

- `call` - outgoing call may be forbidden
- `answer` - incoming call answer may be forbidden
- `hold` - call hold may be forbidden
- `unhold` - call hold may be forbidden
- `transfer` - call transfer may be forbidden
- `sendDTMF` - DTMF sending may be forbidden
- `sendMessage` - message sending may be forbidden
- `sendIMDN` - IMDN sending may be forbidden
- `subscribe` - SIP subscription may be forbidden
- `sendXcapRequest` - XCAP sending may be forbidden
- `publishStream` - stream publishing may be forbidden

- `playStream` - stream playing may be forbidden
- `playHLS` - HLS stream playing may be forbidden
- `playRTSP` - RTSP stream playing may be forbidden
- `OnDataEvent` - data sent may be rejected
- `sendBugReport` - bug report sending may be forbidden

These events may be interrupted, but it is not useful:

- `hangup` - call hangup may be forbidden
- `unPublishStream` - stream unpublishing may be forbidden
- `stopStream` - stream stopping may be forbidden

Events that have already happened and can not be forbidden by backend server

This events can not be cancelled by backend server because the event has already happened on WCS side, and only notification is sent to backend server

- `ConnectionStatusEvent`
- `RegistrationStatusEvent`
- `CallStatusEvent`
- `TransferStatusEvent`
- `MessageStatusEvent`
- `SubscriptionStatusEvent`
- `XcapStatusEvent`
- `StreamStatusEvent`
- `DataStatusEvent`
- `BugReportStatusEvent`
- `StreamKeepAliveEvent`
- `sendStreamEvent`
- `StreamEvent`

Incoming calls

Incoming calls may be interrupted by backend server. WCS notifies caller in this case.

- `OnCallEvent` - incoming call
- `OnMessageEvent` - incoming message

- `OnTransferEvent` - incoming call transfer

If REST method was initiated using REST API

If REST method of any type was initiated using REST API, it should not be declined by backend server.

Actions taken when backend server interaction error is occurred

An interaction error occurs when backend server returns status other than `200 OK`, or some error prevents access to web server. Actions taken depending on `restOnError` field of restClientConfig object and method type, described below:

restOnError	Connect	Direct invocations	Events	Incoming calls
FAIL	1. Decline 2. Return FAIL to a client	1. Log the error 2. Break execution 3. Deliver error to a client in corresponding event	1. Log the error 2. Continue execution 3. Deliver error to a client in special event `ErrorEvent`	1. Log the error 2. Break execution 3. Response to caller with status `403 FORBIDDEN` 4. Deliver error to client in corresponding event
LOG		1. Log the error 2. Continue execution	1. Log the error 2. Continue execution	1. Log the error 2. Continue execution

Actions taken when other error is occurred

When SIP status 4xx, 5xx, 6xx is received, or other error not concerning to REST is occured, an appropriate event will be initiated with `FAILED` status and error description in `info` field. This event will be delivered to web server and then to client according to the rules defined in restClientConfig.

For example, if SIP server returns 403 FORBIDDEN on outgoing call then `CallStatusEvent` event with `status: "FAILED", info:"SIP 403 FORBIDDEN"` and `sipMessageRaw:"the original SIP message"` will be sent to the web server and to the client.

If the error can not be correlated with any of the existing events then ErrorEvent event will be initiated with error description in `info` field.

Let’s review one method of each type (the remaining methods behave similarly within the corresponding type).

Type 1 - the connect method

Type 2 - the direct invoke

Type 3 - the event

Type 4 - the incoming call