Contents

- 1 Requests
- 2 Responses
  - 2.1 1xx Informational Messages
  - 2.2 2xx Successful Responses
  - 2.3 3xx Redirection Responses
  - 2.4 4xx Request Failure Responses
  - 2.5 5xx Server Failure Responses
  - 2.6 6xx Global Failure Responses
  - 2.7 Example: Playback custom error message
  - 2.8 See Also

Requests

- INVITE Indicates that a user is being invited to join a session.
- ACK Confirms that client has received a response to the invite message.
- BYE Terminates a call.
- CANCEL Cancels any Pending Searches.
- OPTIONS Queries for Capabilities.
- REGISTER Registers the Address listed in the header field with a SIP Server.

Responses

1xx Informational Messages

- 100 Trying: Indicates that a request has been initiated by the caller and the called party has yet not been located.
- 180 Ringing: Indicates that the called party has been located and is being notified of the call.
- 181 Call is being forwarded: Indicates that the called party has rerouted the call to another.
- 182 Queued: Indicates that the called party is currently not available, and have put the call in queue.
- 183 Session in Progress

2xx Successful Responses

- 200 OK: Indicates that the request has been successfully processed.

3xx Redirection Responses

- 300 Multiple Choices: Indicates that the address resolved to more than one location.
- 301 Moved permanently: Indicates user is no longer available at this location, an alternate location should be included in the header.
- 302 Moved Temporarily: Indicates that the user in temporarily unavailable, an alternate location should be included in the header.
- 305 Use Proxy: This response indicates that the caller must use a proxy to contact the called party.
- 380 Call is not successful but alternate services are available.

4xx Request Failure Responses

- 400 Bad Request: Indicates the request sent could not be understood.
- 401 Unauthorized Request: Indicates the request requires authorization.
- 402 Payment Required: Indicates payment is required to complete the call.
- 403 Forbidden: Indicates Server has received the request but will not provide the service.
- 404 Not Found: Indicates the server was not found.
- 405 Method Not Allowed: Indicates that the request contains a list of methods that are not allowed.
- 406 Not acceptable: Indicates that the request can not be processed by the client.
- 407 Proxy Authentication Required: Client must first authenticate itself with a proxy.
- 408 Request Timeout: The server could not produce a response before a given time out.
- 409 Conflict: Indicates a conflict with the current state of the resource.
- 410 Gone: Resource is no longer available at the server and no forwarding address was found.
- 411 Length Required: User refuses request without a specified length.
- 412 Request Entity Too Large: Server refuses to process request because URI is too long.
- 415 Unsupported Media: Indicates the format of the body is not supported by the destination endpoint.
- 420 Bad Extension: The server could not understand the protocol extension indicated in the required header.
- 480 Temporarily Unavailable: Indicates that the called party was contacted but was temporarily unavailable.
- 481 Call Leg Transaction Does Not Exist: Indicates that the server was ignoring the request of bye or cancel since there is no matching Invite transaction.
- 482 Loop Detected: (Also, *Request Merged*) Server received a request which has it self in the path.
- 483 Too Many Hops: The server received a request that required more hops than allowed.
- 484 Incomplete Address: The server received a request with an incomplete address.
- 485 Ambiguous: Server received a request in which the called address is ambiguous.
- 486 Busy Here: The called party was contacted but the system was not able to receive any more calls.
- 487 Request Terminated: The calling party canceled the request before the dialog was established with a 200 OK.
- 488 Not Acceptable Here
- 489 Bad Event: See RFC3265
- 491 Request Pending
- 493 Undecipherable
- 494 Security Agreement Required: See RFC3329

5xx Server Failure Responses

- 500 Server Internal Error: Server encountered an unexpected error and could not process the request
- 501 Not Implemented: Server does not support the functions required to complete the request.
- 502 Bad Gateway: Server received an invalid request upstream.
- 503 Service Unavailable: Server has an overload or maintenance problem.
- 504 Gateway Timeout: Server did not receive a timely response from another server.
- 505 Version Not Supported: Server does not support the SIP protocol used in the request.

6xx Global Failure Responses

- 600 Busy Everywhere: Called party is busy and cannot take the call at this time.
- 603 Decline: Called party was contacted but does not want to take part in the call.
- 604 Does Not Exist Anywhere: Called Party does not exist anywhere in the network.
- 606 Not Acceptable: Called party has rejected some part of the call session description as unacceptable.

Example: Playback custom error message

```
<action application="set" data="hangup_on_bridge=true"/>
<action application="set" data="continue_on_fail=true"/>
<action application="bridge" data="..."/>
<action application="lua" data="playerror.lua"/>
```

That'll run the lua script after the bridge, but only if the bridge fails.

```
In the lua script:
  cause = session:getVariable("originate_disposition")
  if (cause == "USER_BUSY") then
     session:streamFile("ivr/ivr-user_busy.wav");
  elseif (cause == "UNALLOCATED_NUMBER") then
     session:streamFile("ivr/ivr-unallocated_number.wav");
  elseif (cause == "NO_USER_RESPONSE" or cause=="NO_ANSWER") then
     session:streamFile("ivr/ivr-no_user_response.wav");
  else
     session:streamFile("ivr/ivr-please_check_number_try_again.wav");
  end
```

See Hangup_Causes for the full list of causes.

Alternatively you can now use transfer_on_fail like this

```
<action application="set" data="hangup_on_bridge=true"/>
<action application="set" data="continue_on_fail=true"/>
<action application="set" data="transfer_on_fail=UNALLOCATED_NUMBER auto_cause xml error"/>
<action application="bridge" data="A..."/>

<action application="bridge" data="B..."/>
```

then have the an "Error" context in your dialplan.

```
<context name="error">
    <extension name="UNALLOCATED_NUMBER" continue="true">
      <condition field="${originate_disposition}" expression="UNALLOCATED_NUMBER" continue="false
        <action application="playback" data="/usr/local/freeswitch/sounds/NotInService.wav"/>
        <action application="hangup" data="NORMAL_CLEARING"/>
      </condition>
    </extension>
</context>
```

See Also

- IANA SIP Parameters