

RCS messages

API: bot/v2/<botId>/messages

Http Method: Post

- RCS Text messages with different type of suggestions

Example: 1

Scenario:

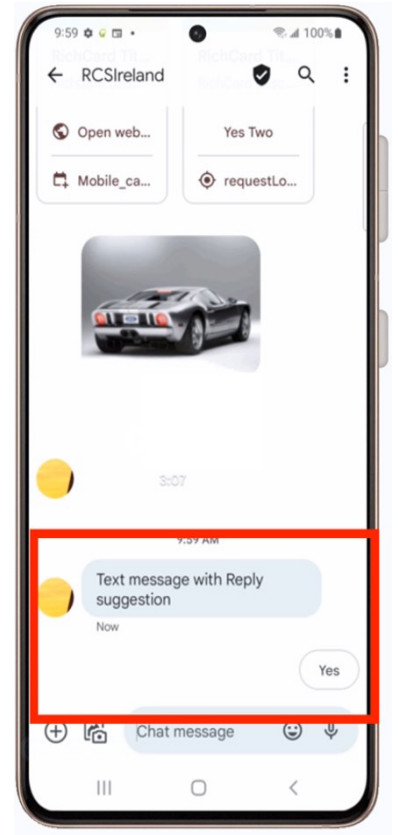
Text message with suggestion type as replay

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with Replay suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "reply": {
            "displayText": "Yes",
            "postback": {
              "data": "set_by_chatbot_reply_yes"
            }
          }
        }
      ]
    }
  },
  "messageContact": {
    "userContact": "+16478850240"
  }
}
```

Response:

```
{
  "RCSMessage": {
    "msgId": "a9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```



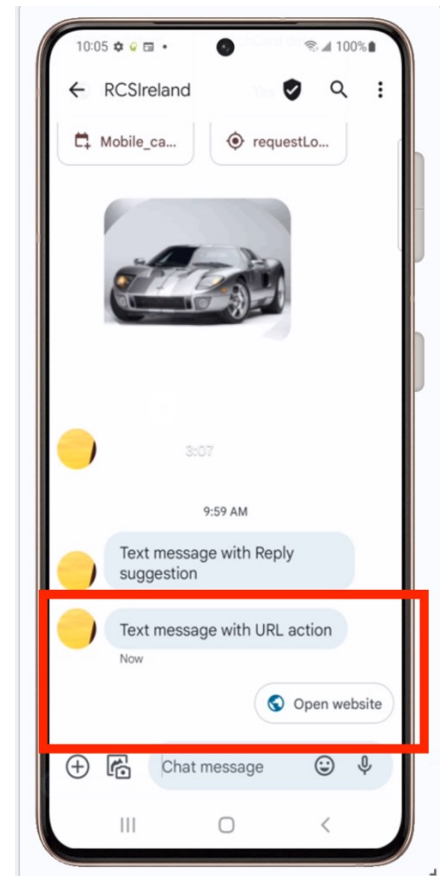
Example: 2

Scenario:

Text message with action type as: UrlAction

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with URL action",
    "suggestedChipList": {
      "suggestions": [
        {
          "action": {
            "urlAction": {
              "openUrl": {
                "url": "https://www.gsma.com"
              }
            }
          },
          "displayText": "Open website",
          "postback": {
            "data": "set_by_chatbot_open_url"
          }
        }
      ]
    }
  },
  "messageContact": {
    "userContact": "+16478850240"
  }
}
```



Response:

```
{
  "RCSMessage": {
    "msgId": "b9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```

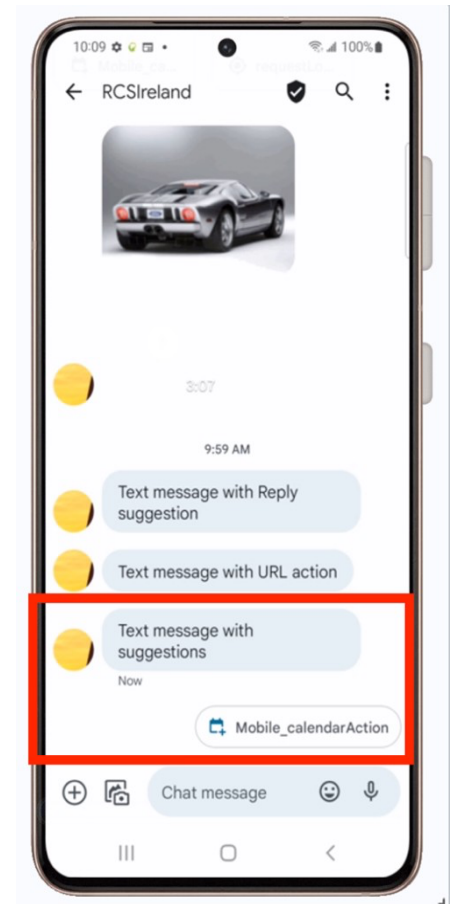
Example: 3

Scenario:

Text message with action type as : calendar Action

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "action": {
            "calendarAction": {
              "createCalendarEvent": {
                "startTime": "2022-03-
14T00:00:00Z",
                "endTime": "2023-03-
14T00:00:00Z",
                "title": "http://www.gsma.com",
                "description":
"http://www.gsma.com"
              }
            },
            "displayText": "Mobile_calendarAction",
            "postback": {
              "data": "e460dd65-e69b-4f86-a5af-
458683248f2d_url_google"
            }
          }
        }
      ]
    }
  },
  "messageContact": {
    "userContact": "+16478850240"
  }
}
```



Response:

```
{  
  "RCSMessage": {  
    "msgId": "c9d0af88-df4e-4ff4-981f-315543056b98",  
    "status": "pending",  
    "timestamp": "2022-11-07T23:58:19.665+05:30"  
  }  
}
```

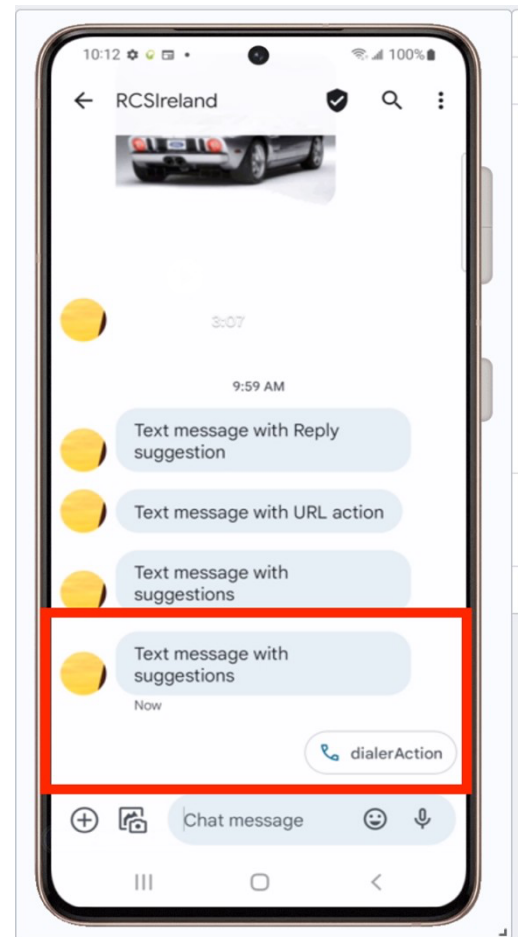
Example: 4

Scenario:

Text message with suggestion action as : dialerAction

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "action": {
            "dialerAction": {
              "dialPhoneNumber": {
                "+1999999999",
                "fallbackUrl":
                "www.webex.com"
              }
            },
            "displayText": "dialerAction",
            "postback": {
              "data":
                "set_by_chatbot_dialerAction_dialPhoneNumber"
            }
          }
        }
      ]
    },
    "messageContact": {
      "userContact": "+16478850240"
    }
  }
}
```



Response:

```
{
  "RCSMessage": {
    "msgId": "d9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```

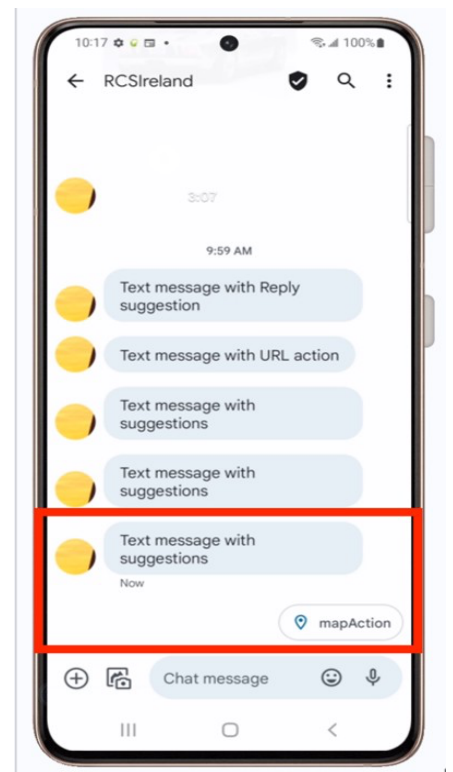
Example: 5

Scenario:

Text message with suggestion action as: mapAction

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "action": {
            "mapAction": {
              "showLocation": {
                "location": {
                  "latitude": 43.649269,
                  "longitude": -79.378423
                }
              }
            }
          },
          "displayText": "mapAction",
          "postback": {
            "data": "set_by_chatbot_mapAction"
          }
        }
      ]
    }
  },
  "messageContact": {
    "userContact": "+16478850240"
  }
}
```



Response:

```
{
  "RCSMessage": {
    "msgId": "e9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```

Example: 6

Scenario:

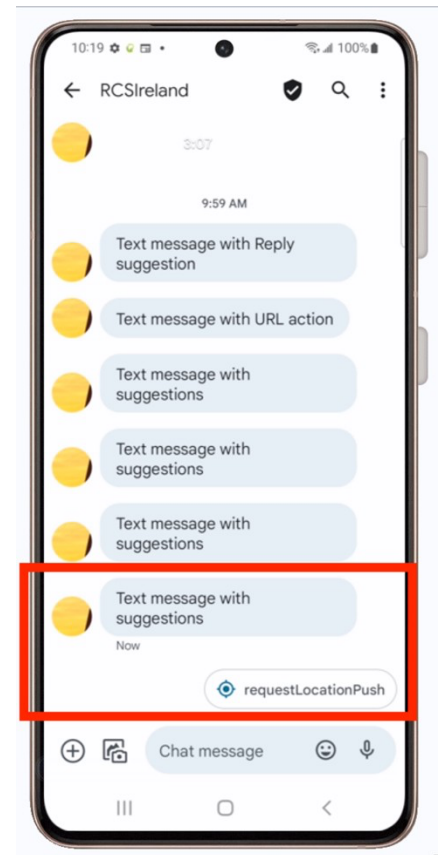
Text message with suggestion as mapAction
(device location)

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "action": {
            "displayText": "requestLocationPush",
            "postback": {
              "data": "requestLocationPushData"
            },
            "mapAction": {
              "requestLocationPush": {}
            }
          }
        }
      ]
    }
  },
  "messageContact": {
    "userContact": "+16478850240"
  }
}
```

Response:

```
{
  "RCSMessage": {
    "msgId": "f9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```



- RCS messages as Rich Card

Example: 1

Request:

```
{
  "RCSMessage": {
    "richcardMessage": {
      "message": {
        "generalPurposeCard": {
          "layout": {
            "cardOrientation": "HORIZONTAL",
            "imageAlignment": "LEFT"
          },
          "content": {
            "media": {
              "mediaUrl":
https://www.edigitalagency.com.au/wp-content/uploads/google-logo-png-transparent-background-large-new.png,
              "height": "MEDIUM_HEIGHT"
            },
            "title": "This is a single rich card.",
            "description": "This is the description of the rich card."
          }
        }
      }
    }
  },
  "messageContact": {
    "userContact": "+16478850240"
  }
}
```



Response:

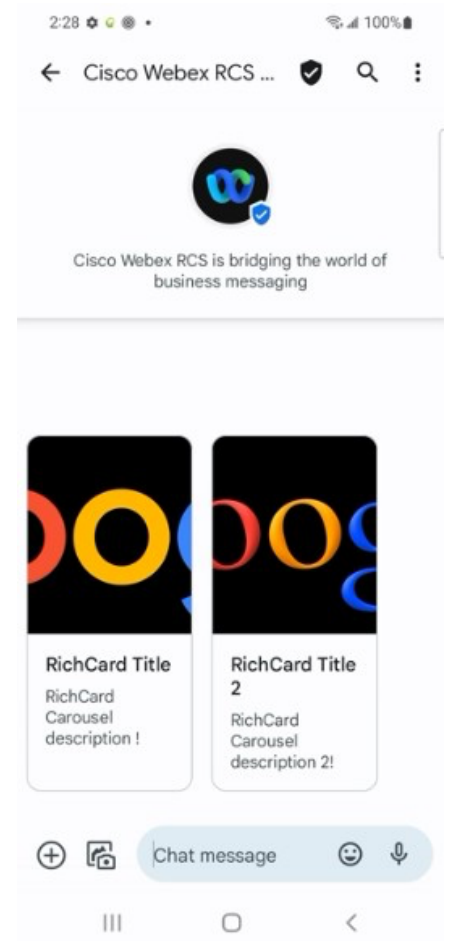
```
{
  "RCSMessage": {
    "msgId": "h9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```


- RCS messages as Rich Card with Carousel

Example: 1

Request:

```
{
  "RCSMessage": {
    "richcardMessage": {
      "message": {
        "generalPurposeCardCarousel": {
          "layout": {
            "cardWidth": "SMALL_WIDTH"
          },
          "content": [
            {
              "media": {
                "height": "MEDIUM_HEIGHT",
                "mediaUrl":
"https://www.edigitalagency.com.au/wp-content/uploads/google-
logo-png-transparent-background-large-new.png",
                "mediaContentType": "image/png",
                "thumbnailUrl":
"https://www.edigitalagency.com.au/wp-content/uploads/google-
logo-png-transparent-background-large-new.png",
                "thumbnailContentType": "image/png"
              },
              "title": "RichCard Title",
              "description": "RichCard description !"
            },
            {
              "media": {
                "height": "MEDIUM_HEIGHT",
                "mediaUrl":
"https://pngimg.com/uploads/google/google_PNG102343.png",
                "mediaContentType": "image/png",
                "thumbnailUrl":
"https://pngimg.com/uploads/google/google_PNG102343.png",
                "thumbnailContentType": "image/png"
              },
              "title": "RichCard Title 2",
              "description": "RichCard description 2!"
            }
          ]
        }
      }
    }
  }
}
```



```
  },  
  "messageContact": {  
    "userContact": "{{phone}}"  
  }  
}
```

Response:

```
{  
  "RCSMessage": {  
    "msgId": "h9d0af88-df4e-4ff4-981f-315543056b98",  
    "status": "pending",  
    "timestamp": "2022-11-07T23:58:19.665+05:30"  
  }  
}
```

- RCS messages as Media Message

Example: 1

Scenario:

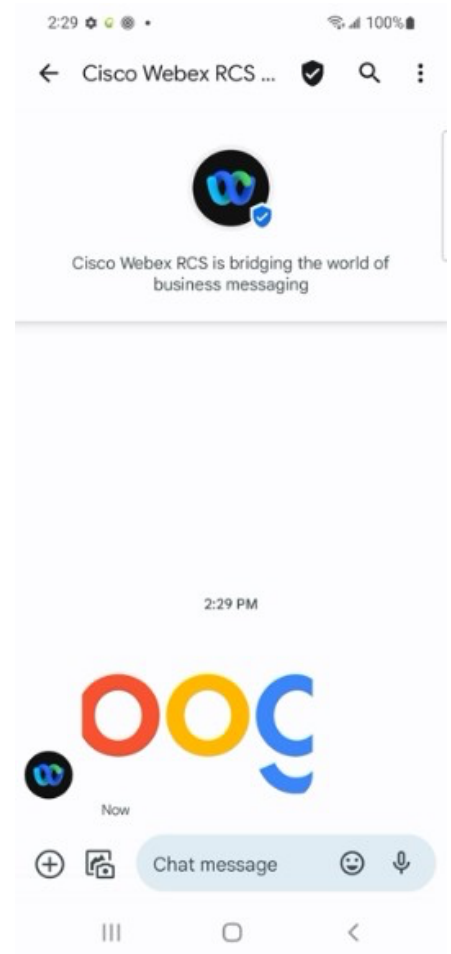
Send file message to user device

Request:

```
{
  "RCSMessage": {
    "fileMessage": {
      "fileUrl": "https://www.edigitalagency.com.au/wp-content/uploads/google-logo-png-transparent-background-large-new.png"
    }
  },
  "messageContact": {
    "userContact": "+13399276309"
  }
}
```

Response:

```
{
  "RCSMessage": {
    "msgId": "j9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```



Example: 2

Scenario:

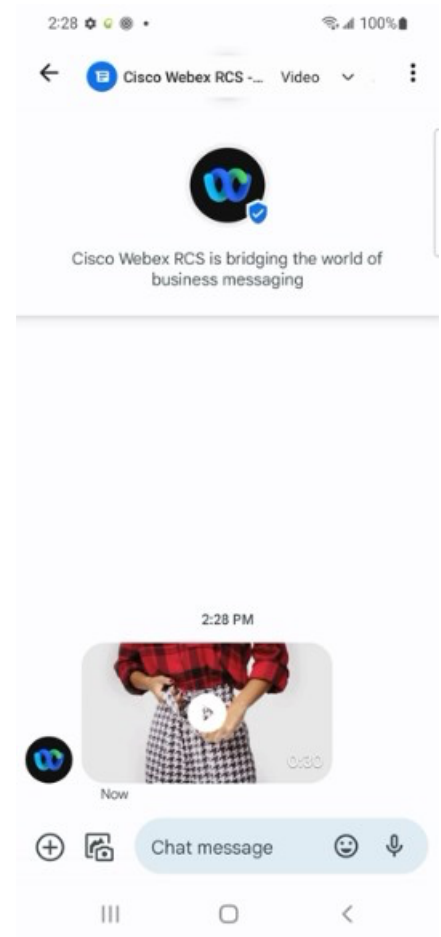
Send audio message to user device.

Request:

```
{
  "RCSMessage": {
    "audioMessage": {
      "fileUrl": "https://3crcs-cms-prod.s3.amazonaws.com/rcs-cms/f2c2aa52-82d1-4301-a032-ead248e076de/hero-640x360-2mbps.mp4"
    }
  },
  "messageContact": {
    "userContact": "+13399276309"
  }
}
```

Response:

```
{
  "RCSMessage": {
    "msgId": "j9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```



- RCS message as Typing

Request:

```
{
  "RCSMessage": {
    "isTyping": "active",
    "callbackData": "callMeBack"
  },
  "messageContact": {
    "correlationId": "testCorrelationId",
    "userContact": "+13399276309"
  }
}
```

Response:

```
{
  "RCSMessage": {
    "msgId": "k9d0af88-df4e-4ff4-981f-315543056b98",
    "status": "pending",
    "timestamp": "2022-11-07T23:58:19.665+05:30"
  }
}
```

Scenario:

When incoming request is rejected due to bad json.

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with Replay suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "reply": {
            "displayText": ["Yes"],
            "postback": {
              "data": "set_by_chatbot_reply_yes"
            }
          }
        }
      ]
    }
  },
  "messageContact": {
    "correlationId": "testCorrelationId",
    "userContact": "+13399276309"
  }
}
```

Response:

```
{
  "Reason": {
    "text": "Invalid JSON",
    "code": 7000
  }
}
```

Scenario:

When user request is missing mandatory attribute.

Request:

```
{
  "RCSMessage": {
    "textMessage": "Text message with Replay suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "reply": {
            "displayText": "Yes",
            "postback": {
              "data": "set_by_chatbot_reply_yes"
            }
          }
        }
      ]
    }
  },
  "messageContact": {
  }
}
```

Response:

```
{
  "Reason": {
    "text": "Mandatory parameters missing - to",
    "code": 7003
  }
}
```

Scenario:

When incoming request do not have msisdn information.

Request:

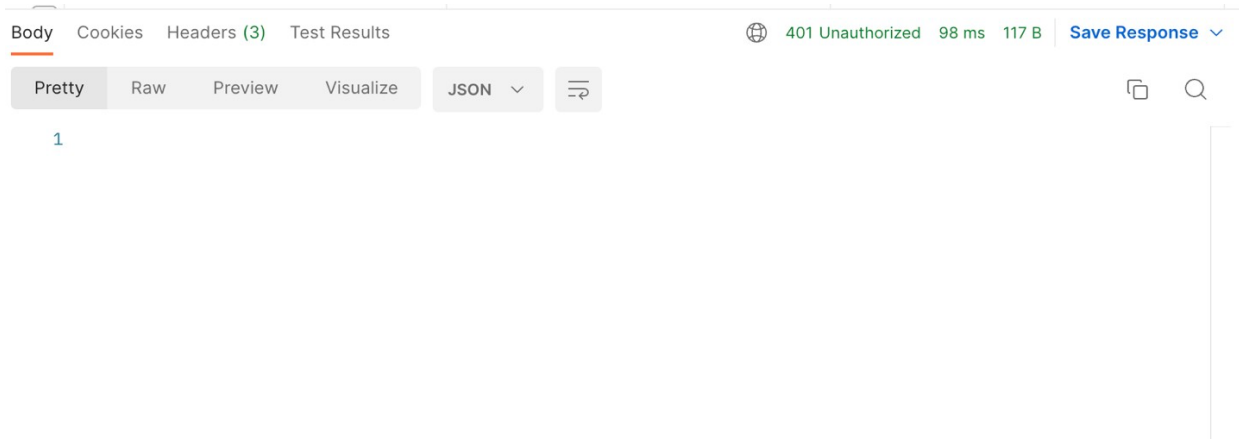
```
{
  "RCSMessage": {
    "textMessage": "Text message with Replay suggestions",
    "suggestedChipList": {
      "suggestions": [
        {
          "reply": {
            "displayText": "Yes",
            "postback": {
              "data": "set_by_chatbot_reply_yes"
            }
          }
        }
      ]
    }
  },
  "messageContact": {
    "userContact": ""
  }
}
```

Response:

```
{
  "Reason": {
    "text": "Invalid parameter - destination",
    "code": 7004
  }
}
```


Scenario:

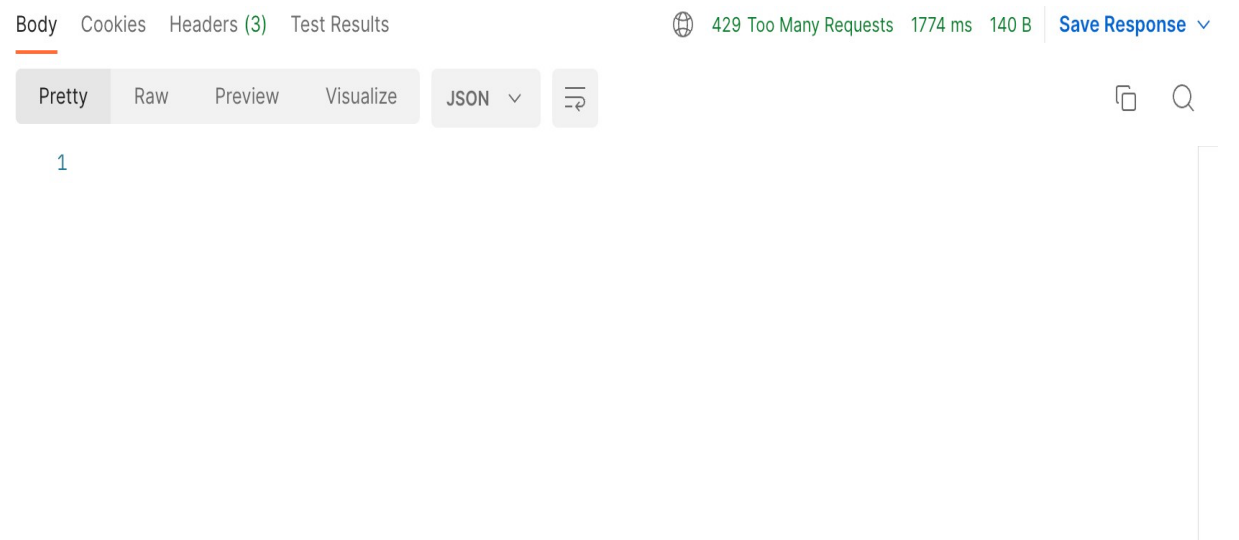
When incoming request do not have authorization information or invalid authorization key provided.



Scenario:

When user is making request faster than expected rate. (too many request)

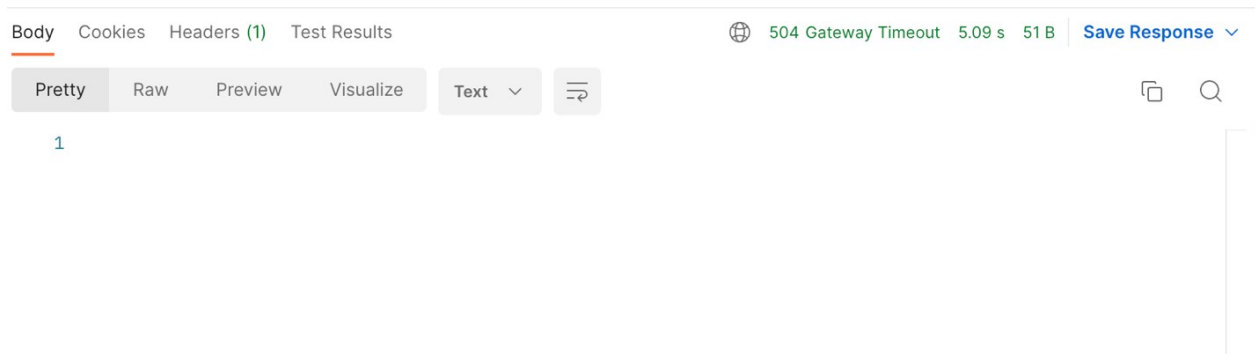
Response:



Scenario:

When gsma proxy is not getting response back in configure time from its internal system (connect system) then it will be timeout at proxy level which marked as gateway timeout.

Response:



Scenario:

When gsma proxy received timeout response from its internal system then response will be marked as **request time out with http code as: 408**

Capability lookup

Api: bot/v1/<botId>/contactCapabilities?userContact=<msisdn>

Http Method: Get

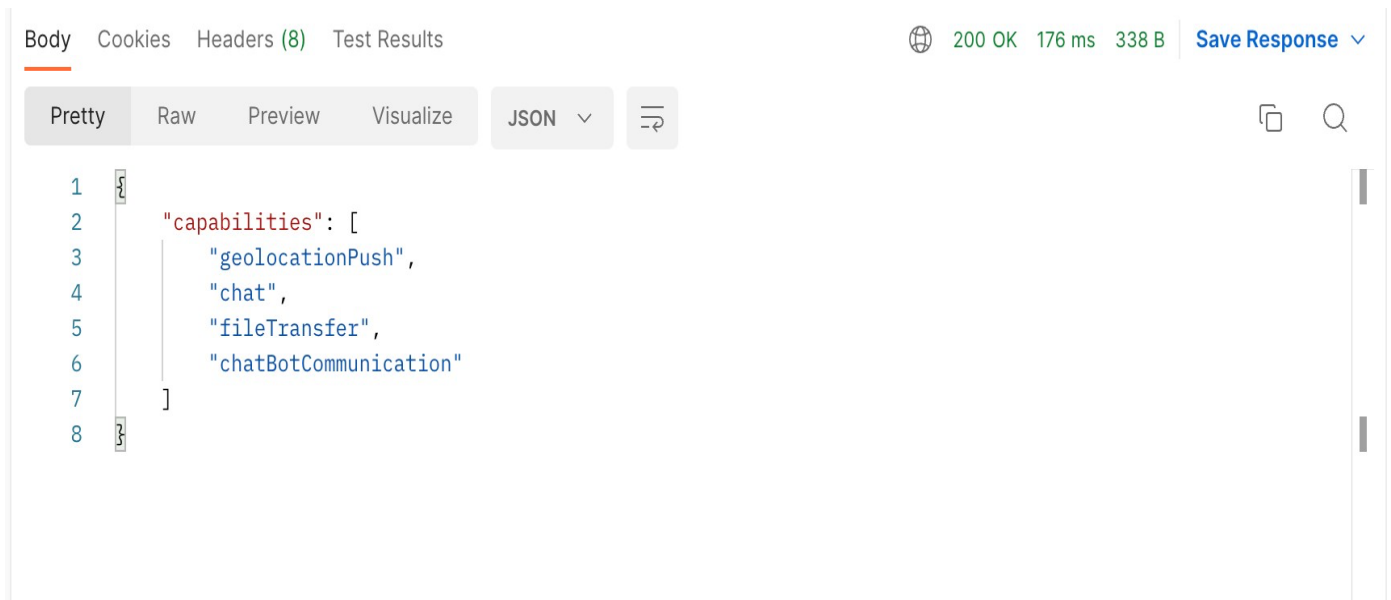
Scenario:

When user made a capability lookup request with valid bot id and msisdn is RCS capable.

Request:

bot/v1/a_159840733651579260/contactCapabilities?userContact=+13399276309

Response:



The screenshot displays the 'Body' tab of a web browser's developer tools. The response is a JSON object with a 'capabilities' array. The status bar at the top indicates a 200 OK response, 176 ms latency, and 338 B of data. The JSON content is as follows:

```
{
  "capabilities": [
    "geolocationPush",
    "chat",
    "fileTransfer",
    "chatBotCommunication"
  ]
}
```

Scenario:

When user made a capability lookup request with invalid botId.

Request:

bot/v1/a_159840733651579261/contactCapabilities?userContact=+13399276309

Response:



The screenshot shows a REST client interface with the following details:

- Body:** Selected tab.
- Headers (3):** Tab is active.
- Test Results:** Tab is active.
- Status:** 404 Not Found, 88 ms, 163 B.
- Save Response:** Button with a dropdown arrow.
- Format:** JSON (selected), Pretty (selected), Raw, Preview, Visualize.
- JSON Content:**

```
1 {
2   "reason": {
3     "code": 7104,
4     "text": "Invalid app ID"
5   }
6 }
```

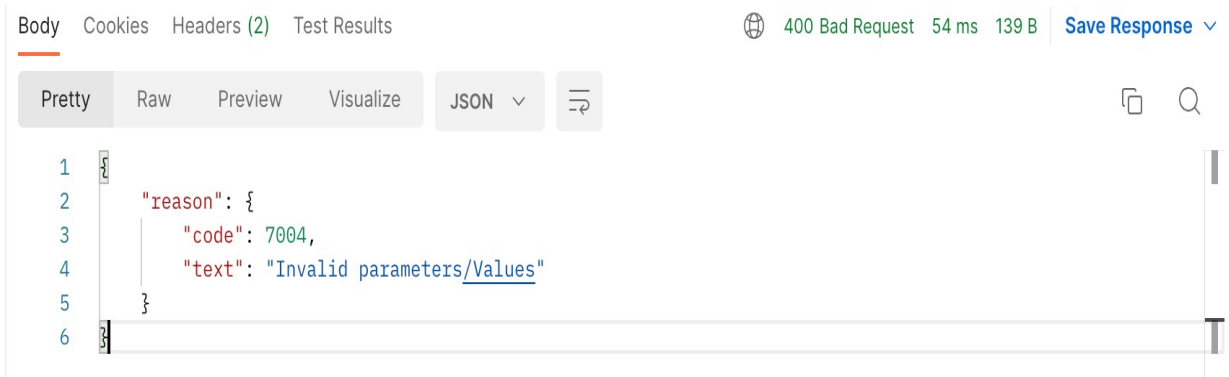
Scenario:

When user request does not have query param value(userContact) information.

Request:

v1/a_159840733651579261/contactCapabilities

Response:



The screenshot shows a REST client interface with the following details:

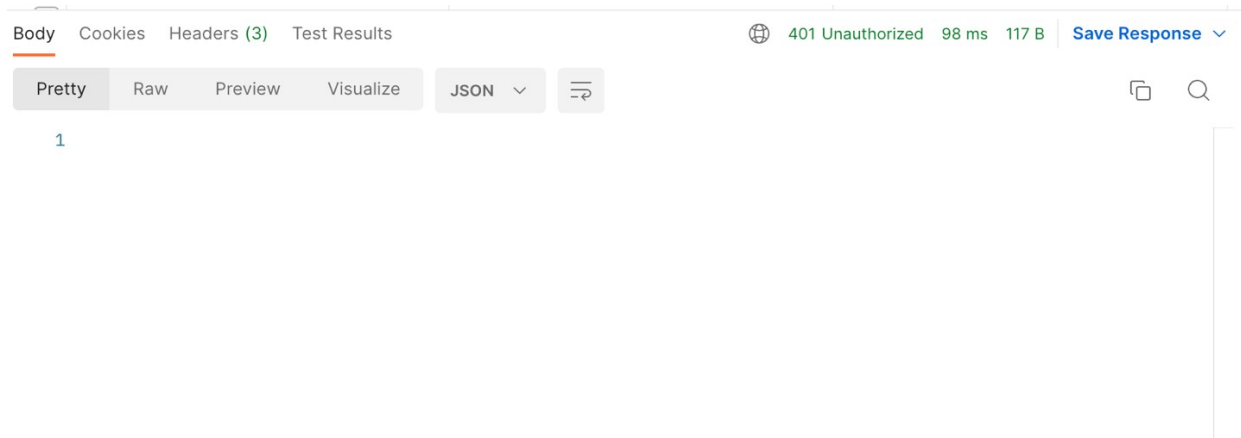
- Body:** Selected tab.
- Headers (2):** Tab is active.
- Test Results:** Tab is active.
- Status:** 400 Bad Request, 54 ms, 139 B.
- Save Response:** Button with a dropdown arrow.
- Format:** JSON (selected), Pretty (selected), Raw, Preview, Visualize.
- JSON Content:**

```
1 {
2   "reason": {
3     "code": 7004,
4     "text": "Invalid parameters/Values"
5   }
6 }
```

Scenario:

When incoming request do not have authorization information or provided information is invalid.

Response:



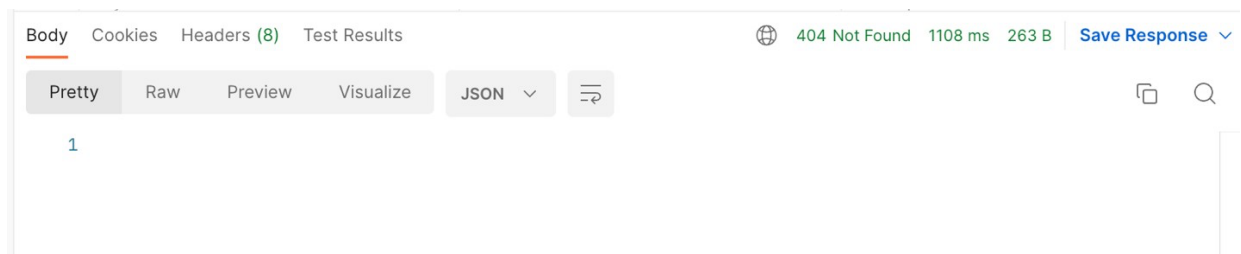
Scenario:

When provided msisdn in not RCS capable.

Request:

bot/v1/a_159840733651579260/contactCapabilities?userContact=+13399276308

Response:



Update Message status

Api: bot/v2/<botId>/messages/<msgId>/status

Http Method: Put

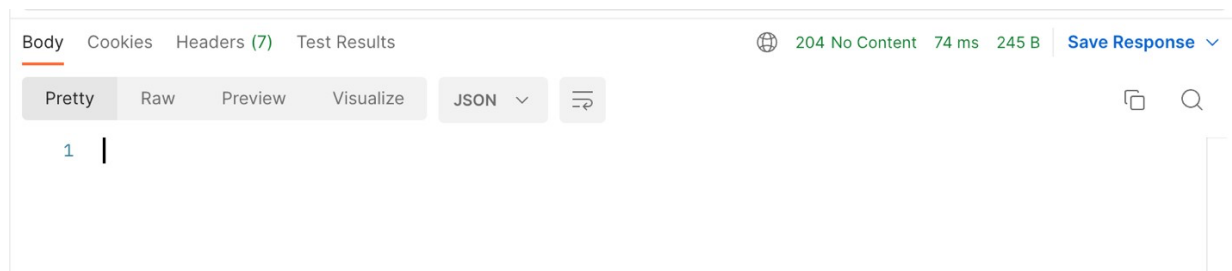
Scenario:

Valid botId with valid msgId.

Request:

```
{
  "RCSMessage": {
    "status": "displayed"
  },
  "messageContact": {
    "userContact": "+13399276309"
  }
}
```

Response:



Scenario:

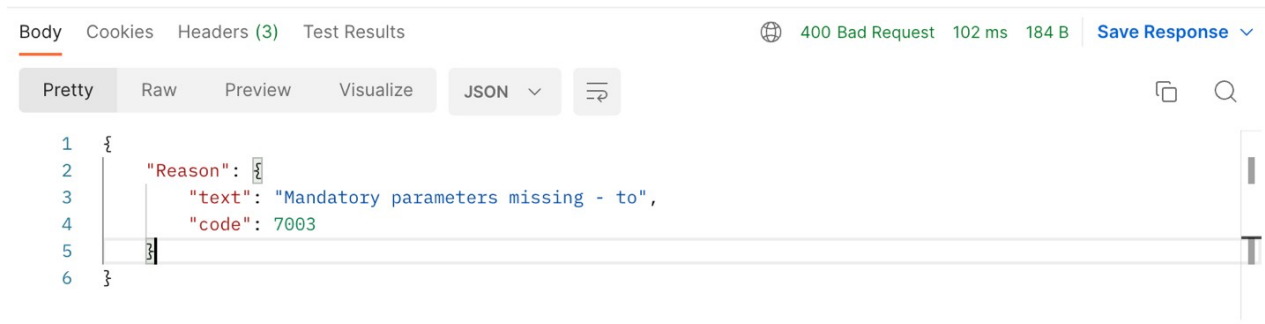
When incoming request is missing mandatory information.

Example: 1

Request:

```
{
  "RCSTMessage": {
    "status": "displayed"
  },
  "messageContact": {
  }
}
```

Response:



The screenshot shows a REST client interface with the following details:

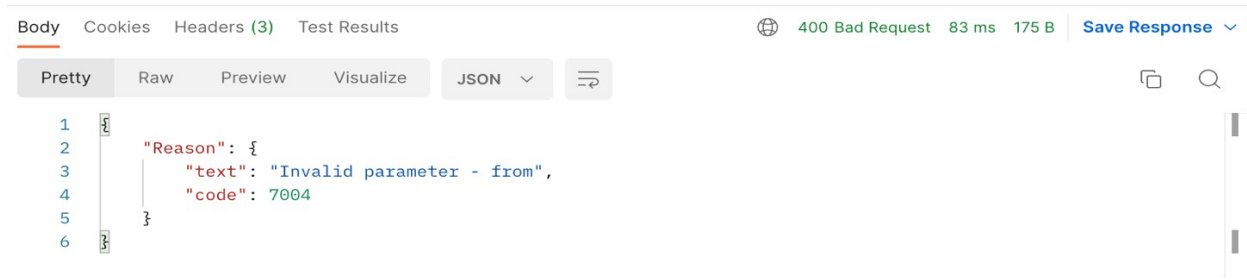
- Body** tab selected.
- Headers (3)** tab.
- Test Results** tab.
- 400 Bad Request** status.
- 102 ms** response time.
- 184 B** response size.
- Save Response** button.
- Pretty** view selected.
- JSON** format selected.
- Response Body:**

```
1 {
2   "Reason": {
3     "text": "Mandatory parameters missing - to",
4     "code": 7003
5   }
6 }
```

Example: 2

Provided bot id is invalid.

Response:



The screenshot shows a REST client interface with the following details:

- Body** tab selected.
- Headers (3)** tab.
- Test Results** tab.
- 400 Bad Request** status.
- 83 ms** response time.
- 175 B** response size.
- Save Response** button.
- Pretty** view selected.
- JSON** format selected.
- Response Body:**

```
1 {
2   "Reason": {
3     "text": "Invalid parameter - from",
4     "code": 7004
5   }
6 }
```

Scenario:

Input request and bot id is valid but msgId is not provided.

Request:

bot/v2/a_159840733651579261/messages//status

Response:

The screenshot shows a web browser interface with a status bar at the top indicating a 405 Method Not Allowed response. The status bar includes a globe icon, the text "405 Method Not Allowed", the response time "53 ms", the response size "54 B", and a "Save Response" button with a dropdown arrow. Below the status bar, there are tabs for "Body", "Cookies", "Headers (1)", and "Test Results". The "Body" tab is selected and highlighted with an orange underline. Below the tabs, there are buttons for "Pretty", "Raw", "Preview", and "Visualize", followed by a "Text" button with a dropdown arrow and a refresh icon. The main content area is empty, and a small "1" is visible in the bottom left corner.

Scenario:

User request do not have authorization information or provided information is invalid.

Response:

The screenshot shows a web browser interface with a status bar at the top indicating a 401 Unauthorized response. The status bar includes a globe icon, the text "401 Unauthorized", the response time "98 ms", the response size "117 B", and a "Save Response" button with a dropdown arrow. Below the status bar, there are tabs for "Body", "Cookies", "Headers (3)", and "Test Results". The "Body" tab is selected and highlighted with an orange underline. Below the tabs, there are buttons for "Pretty", "Raw", "Preview", and "Visualize", followed by a "JSON" button with a dropdown arrow and a refresh icon. The main content area is empty, and a small "1" is visible in the bottom left corner.