

## **O2 RCS BUSINESS MESSAGING**



Start guide



# CONTENTS

Contents .....	2
1. About this document .....	3
1.1. Revision history .....	3
2. Chatbot Agent credentials .....	4
3. Understand Basic Terminology .....	4
3.1. Introduction .....	4
3.2. Basic terminology .....	4
3.3. Supported media types .....	4
3.4. Card template .....	4
3.5. Carousel Templates .....	6
3.6. Suggested Actions Template .....	7
3.7. Limitations .....	8
3.7.1. General limitations .....	8
3.7.2. Element limitations .....	8
3.7.3. Character limitations .....	8
3.7.4. Sizing .....	8
3.7.5. Error conditions .....	8
3.8. Tips for RCS messages .....	9
3.8.1. Start a new line .....	9
3.8.2. Emojis .....	10
3.8.3. Truncation .....	10
4. Chatbot interaction with RBM platform .....	12
4.1. Access token .....	12
4.1.1. Request token - JavaScript .....	12
4.1.2. Request token - Python .....	12
4.1.3. Responses .....	13
4.1.4. Usage of the access token .....	13
4.2. Bot sending message to User .....	14
4.2.1. Text Message .....	14
4.2.2. Suggested ChipList .....	15
4.2.3. Rich Card with Suggestions .....	16
4.2.4. Rich Card Carousel .....	18
4.2.5. URL Action .....	20
4.2.6. Map Action .....	21
4.2.7. Calendar Action .....	22
4.2.8. Dialer Action .....	23
4.2.9. Send Location .....	24
4.3. User Sending Messaging to Bot .....	26
4.3.1. Text .....	26



4.3.2. Suggested Response..... 26  
4.3.3. User’s Location..... 27  
4.3.4. File sharing ..... 27  
4.3.5. Events (DELIVERED/READ/IS TYPING)..... 28  
5. References ..... 28

## 1. ABOUT THIS DOCUMENT

This document provides basic information about RCS business messages. And shows examples of API commands for various types of RCS messages that can be send via O2 RCS business platform. Link for full API RCS documentation is in section [5.REFERENCES](#)

### 1.1. Revision history

Version	Date	Author	Notes
Version 0.91	14. 10. 2022	Tomáš Zvolský	Web links in references not ready
Version 1.0	23. 2. 2023	Tomáš Zvolský	References update
Version 1.1	8. 12. 2025	Tomáš Zvolský	Refreshed code examples to align with the latest API.



## 2. CHATBOT AGENT CREDENTIALS

Register to [RMB Administration console](#) and create your [Chatbot agent](#)

## 3. UNDERSTAND BASIC TERMINOLOGY

### 3.1. Introduction

Rich Messaging allows you to push something more rich and interactive than just plain text. A single request can include text, a rich card, an image, or a video as well as suggested replies and suggested actions.

### 3.2. Basic terminology

*Agent ID / Client ID*

This is the ID of your chatbot agent when the bot is on boarded. E.g.: RCS\_test\_-\_tomas\_bot\_agent

*JSON Web Token*

Private [key to access](#) the API as your agent

*User MSISDN*

(Mobile Station Integrated Services Digital Network) is the phone number which identifies a device

### 3.3. Supported media types

- .jpg, .png, .mp4, .gif... [More details](#):

### 3.4. Card template

The RCS Business Messaging card has 2 template orientation options:

1. Vertical card
2. Horizontal card

**Vertical rich cards** display horizontal media at the top of the card. Horizontal media should have an aspect ratio of 2:1, 16:9, or 7:3.

When you send media to a user, you should be respectful of the user's resources. When horizontal media has a 4:3 ratio, the optimal resolution for the media is 960x720 px with a maximum recommended file size of 2 MB for images and 10 MB for video. The optimal resolution for the media's thumbnail is 770x335 px with a recommended file size of 40 kB and a recommended maximum size of 100 kB.

**Horizontal rich cards** display vertical media on the left or right side of the card. Vertical media should have an aspect ratio of 3:4.

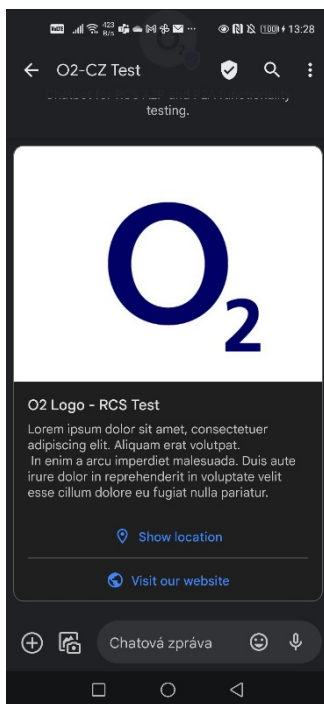
When you send media to a user, you should be respectful of the user's resources. When vertical media has a 3:4 ratio, the optimal resolution for the media is 768x1024 px with a maximum recommended file size of 2 MB for images and 10 MB for video. The optimal resolution for the media's thumbnail is 250x330 px with a recommended file size of 40 kB and a recommended maximum size of 100 kB.



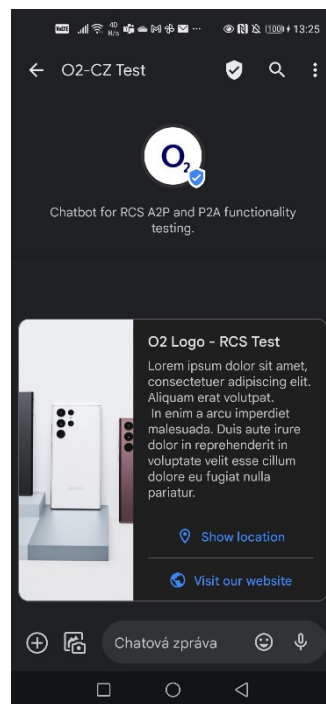
**Both cards contain the following elements:**

- Title text
- Subtitle text (including emojis)
- Image/video
- Up to 4 buttons (RCS limitation) with the following actions:
  - Url action
  - Publish text click action
  - Location action
  - Call action
  - Calendar action
- Up to 11 suggestion actions

*Vertical card example image:*



*Horizontal card example image:*



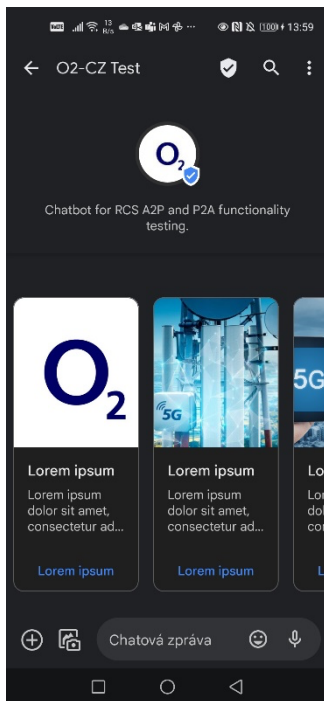
### 3.5. Carousel Templates

The RCS carousel is a horizontally scrollable carousel of up to 10 vertical rich cards. Each card can include the following elements:

- Title text
- Subtitle text
- Image (which will be placed above the title, subtitle and buttons) - image is not required to be placed in a card
- Up to 4 buttons (RCS limitation) with the following actions:
  - Url action
  - Publish text click action
  - Location action
  - Call action
  - Calendar action
- Up to 11 suggestion actions

When the media has a 4:3 aspect ratio, the optimal resolution for the media is 960x720 px with a maximum file size of 1 MB for images and 5 MB for video. The optimal resolution for the media's thumbnail is 605x452 px with a recommended file size of 40 kB and a recommended maximum size of 100 kB.

*Carousel example image:*





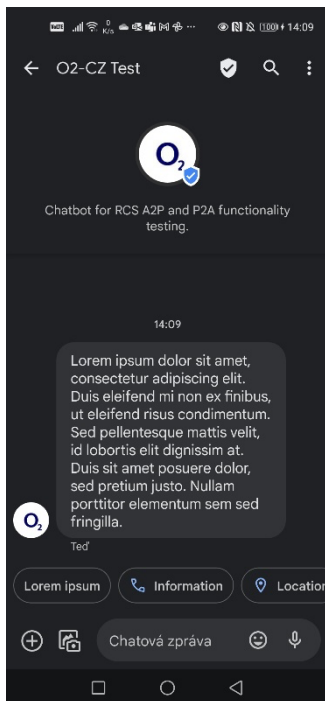
### 3.6. Suggested Actions Template

Quick Replies for RCS Business Messaging include the question/statement text and a set of up to 11 replies that each contain:

- Url action
- Publish text click action
- Location action
- Call action
- Calendar action

Suggested actions allow an agent to hook into native device actions and provide a tightly integrated experience for the user. When relevant, suggested actions can make calling customer support or finding a location on the map easy, but don't overwhelm users with options. Only provide actions that are related to the most recent message, and only provide as many actions as necessary. Limit the number of suggested actions and suggested replies to what is useful and useable for the user in given the context.

*Suggested actions example image:*





## 3.7. Limitations

### 3.7.1. General limitations

- A rich card can contain any or all of the supported elements: image, title, subtitle, buttons. However, it must contain at least an image or title to be valid.
- RCS Business Messaging allows consumers to perform various actions using button actions — share user location, dial phone number, add to calendar, payment request

### 3.7.2. Element limitations

- Cards have a maximum of 4 buttons
- Carousels have a maximum of 10 scrollable cards

### 3.7.3. Character limitations

- Button text can contain up to 25 characters
- Google doesn't have a hard limitation on number of characters within title or subtitle, but generally recommend to be concise

### 3.7.4. Sizing

- RCS Card Images sizes — RCS card images are available in three heights: short, medium, tall.
- RCS Carousel Images sizes RCS card images are available in three heights: short, medium, tall with medium or small width
  - Google recommends the image file size should not exceed 2MB.
  - If the image size added doesn't fit the dimensions of the card's height, the image preview is chosen by zooming and cropping the image.

### 3.7.5. Error conditions

- Agent or bot will receive an error when trying to send RCS Structured Content elements with the following character limitations:
  - If button text exceeds 25 characters
  - If structured content map element is applied



### 3.8. Tips for RCS messages

#### 3.8.1. Start a new line

- \n – start a new line of text

```
---BEGIN Message---  
... "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. \nAliquam erat volutpat. \nIn enim a arcu  
imperdiet malesuada. \nDuis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla  
pariatur. \n\nFusce tellus. \nEtiam sapien elit, consequat eget, tristique non, venenatis quis, ante."  
...  
---END Message---
```

Description without \n in text:

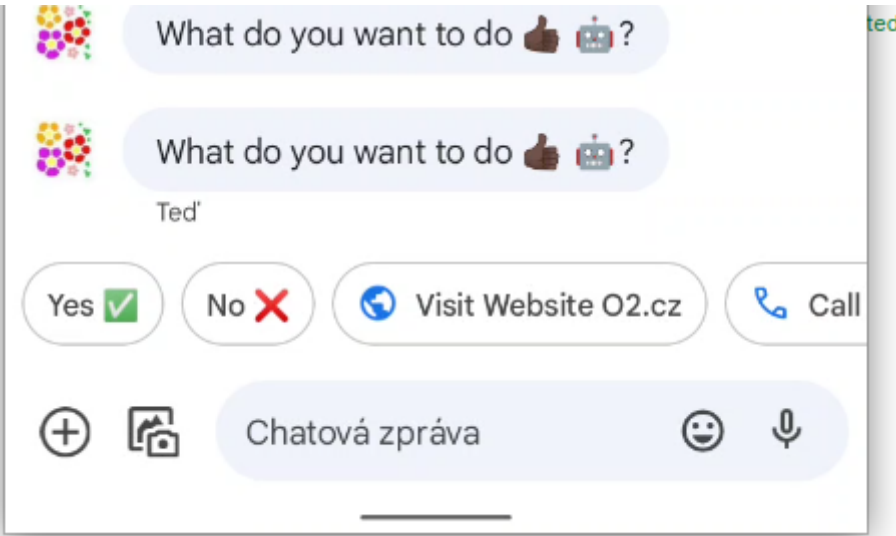


Description where \n was used for new line:



### 3.8.2. Emojis

- Google emojis can be used - [Google Emoji List](#)



### 3.8.3. Truncation

Much like rich cards, many factors (such as screen resolution, pixel density, and user preferences) affect how cards appear to end users. In a carousel, however, the height of the first few cards defines the height of all the cards in the carousel, and card height affects title, description, and suggestion truncation.

If a device can't display all elements of a card because of display constraints or card height, RBM truncates the card until it can display on the device, using the following logic:

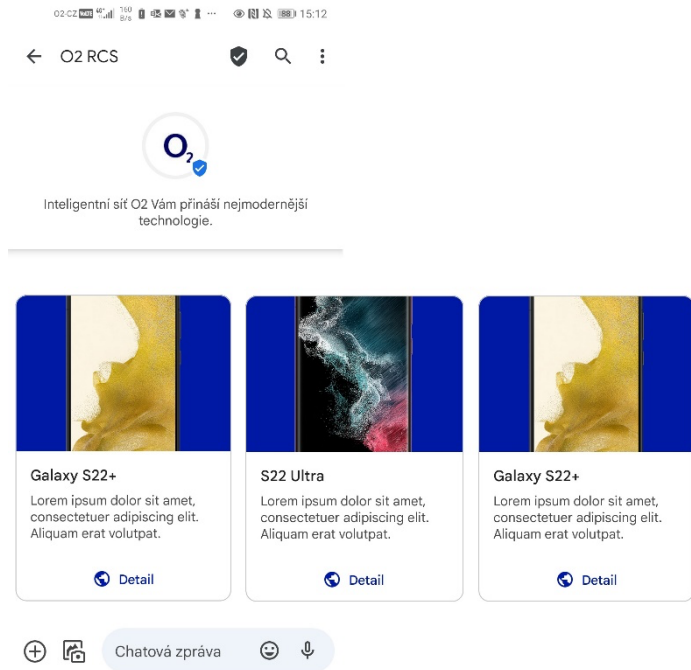
1. Reduce the description to one line.
2. Reduce the title to one line.
3. Omit suggestions that don't fit in the card, starting from the end of the defined list.
4. Omit the description.
5. Omit the title.

To avoid truncation, keep titles and descriptions as short as possible. For tall media, use either a title and description or one suggestion. For medium media, use up to two suggestions. For short media, use up to three suggestions. To fit four suggestions, don't include media in the card.

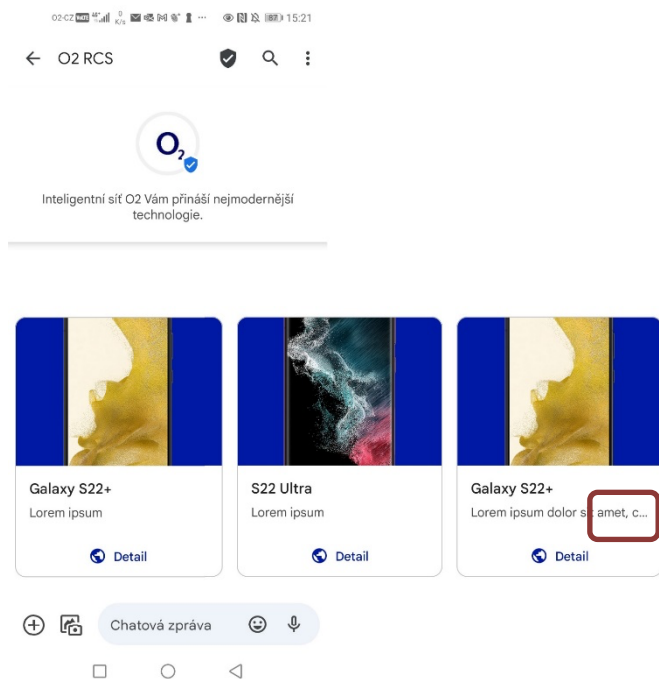
Keep cards roughly equivalent in terms of content sizing and length, and if necessary, front-load the carousel with larger cards to avoid truncation in following cards.



*Description without truncation:*



*Third card with truncation (because first and second card have short description):*





## 4. CHATBOT INTERACTION WITH RBM PLATFORM

### 4.1. Access token

#### 4.1.1. Request token - JavaScript

Prerequisites:

```
1. npm install google-auth-library
```

```
1. const { GoogleAuth } = require('google-auth-library');
2.
3. async function getRCSToken() {
4.   const auth = new GoogleAuth({
5.     keyFile: './service-account.json',
6.     scopes: [
7.       'https://www.googleapis.com/auth/businesscommunications',
8.       'https://www.googleapis.com/auth/businessmessages'
9.     ]
10.  });
11.
12.   const client = await auth.getClient();
13.   const tokenResponse = await client.getAccessToken();
14.   return tokenResponse.token;
15. }
16.
17. // Použití:
18. const token = await getRCSToken();
```

Token can be used in HTTP header  
Authorization: Bearer YOUR\_TOKEN\_HERE  
Token is valid ~1 hour

#### 4.1.2. Request token - Python

Prerequisites:

```
1. pip install google-auth
```

```
1. from google.oauth2 import service_account
2. from google.auth.transport.requests import Request
3.
4. def get_rcs_token():
5.   credentials = service_account.Credentials.from_service_account_file(
6.     'service-account.json',
7.     scopes=[
8.       'https://www.googleapis.com/auth/businesscommunications',
9.       'https://www.googleapis.com/auth/businessmessages'
10.    ]
11.   )
12.   credentials.refresh(Request())
13.   return credentials.token
14.
15. # Použití:
16. token = get_rcs_token()
```

Token can be used in HTTP header  
Authorization: Bearer YOUR\_TOKEN\_HERE  
Token is valid ~1 hour



### 4.1.3. Responses

#### RBM platform return 200 OK

```
1. {
2.   "access_token": "ya29.a0AfH6SMC..._very_long_token_string...",
3.   "expires_in": 3599,
4.   "token_type": "Bearer"
5. }
6.
```

access\_token - token(string)  
expires\_in – validity in second (usually 3599 = ~1 hour)  
token\_type – always parameter "Bearer"

#### RBM return 400 Bad Request

```
1. {
2.   "error": "invalid_grant",
3.   "error_description": "Invalid JWT: Token must be a short-lived token"
4. }
5.
```

#### 403 Forbidden

```
1. {
2.   "error": "access_denied",
3.   "error_description": "Requested client not authorized"
4. }
5.
```

### 4.1.4. Usage of the access token

ChatBot at startup should send the following REST API to get the token first time using one of the options provided below.

#### How to send access token in the message

```
1. POST https://europe-
rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4. {
5.   'contentMessage': {
6.     'richCard': {
7.       'standaloneCard': {
8.         *****
9.       }
10.    }
11. }
```



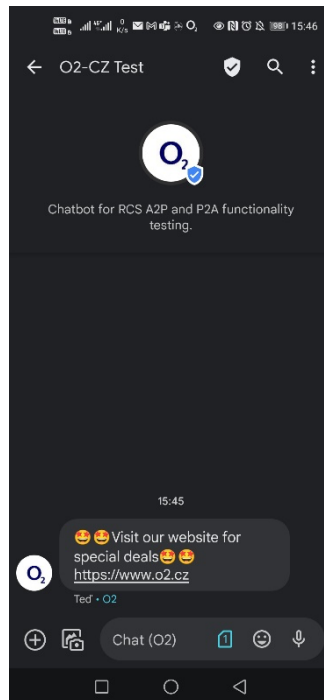
## 4.2. Bot sending message to User

### 4.2.1. Text Message

Bot can use the below as a reference to send text message to user

```
1. POST https://europe-
rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   'contentMessage': {
7.     'text': '😁😁Visit our website for special deals😁😁\nhttps://www.o2.cz'
8.   }
9. }
```

Display on the OTT will be like below:



#### 4.2.2. Suggested ChipList

Bot can use the below as a reference to send Suggested ChipList to user.

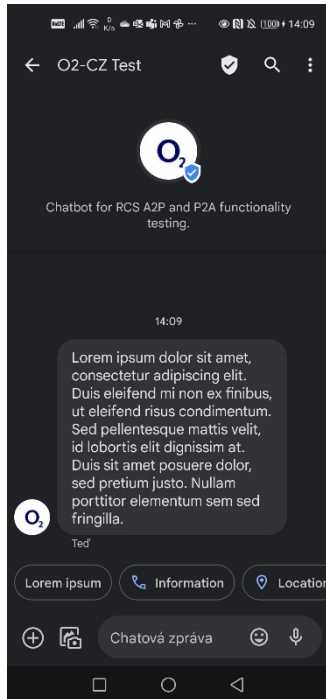
This is the Suggested Chip List JSON object following UP 2.0 specification, a.k.a. the "suggestions" object defined in 3.6.10.3 of RCC.07 Specification.

*Suggested Chip List Request:*

```
1. POST https://europe-
rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "text": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis eleifend mi non ex finibus, ut
eleifend risus condimentum. Sed pellentesque mattis velit, id lobortis elit dignissim at. Duis sit amet posuere
dolor, sed pretium justo. Nullam porttitor elementum sem sed fringilla.",
8.     "suggestions": [
9.       {
10.        "reply": {
11.          "text": "Lorem ipsum",
12.          "postbackData": "NOREPLY"
13.        }
14.      },
15.      {
16.        "action": {
17.          "text": "Information",
18.          "postbackData": "NOREPLY",
19.          "dialAction": {
20.            "phoneNumber": "+420800020202"
21.          }
22.        }
23.      },
24.      {
25.        "action": {
26.          "text": "Location",
27.          "postbackData": "NOREPLY",
28.          "openUrlAction": {
29.            "url": "https://maps.google.com/?q=O2+CZ"
30.          }
31.        }
32.      }
33.    ]
34.  }
35. }
```



Display on the OTT will be like below:



### 4.2.3. Rich Card with Suggestions

Bot can use the below as a reference to send Suggested ChipList to user.

This is the Suggested Chip List JSON object following UP 2.0 specification, a.k.a. the "suggestions" object defined in 3.6.10.3 of RCC.07 Specification.

```
1. POST https://europe-
rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "richCard": {
8.       "standaloneCard": {
9.         "cardContent": {
10.          "title": "O2 Logo - RCS Test",
11.          "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat.\n In
enim a arcu imperdiet malesuada. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariat.",
12.          "media": {
13.            "height": "TALL",
14.            "contentInfo": {
15.              "fileUrl": "https://www.o2.cz/_pub/a6/fb/2b/237540_504860_02_RGB.jpg",
16.              "forceRefresh": false
17.            }
18.          },
19.          "suggestions": [
20.            {
21.              "action": {
22.                "text": "Visit our Web",
23.                "postbackData": "NOREPLY",
24.                "openUrlAction": {
25.                  "url": "https://www.o2.cz/"
26.                }
27.              }
28.            }
29.          ]
30.        }
31.      }
32.    }
33.  }
```



```
28.     }
29.   ]
30. },
31.   "cardOrientation": "VERTICAL"
32. }
33. },
34. "suggestions": [
35.   {
36.     "reply": {
37.       "text": "Lorem ipsum",
38.       "postbackData": "NOREPLY"
39.     }
40.   },
41.   {
42.     "action": {
43.       "text": "Information",
44.       "postbackData": "NOREPLY",
45.       "dialAction": {
46.         "phoneNumber": "+420800020202"
47.       }
48.     }
49.   },
50.   {
51.     "action": {
52.       "text": "Location",
53.       "postbackData": "NOREPLY",
54.       "openUrlAction": {
55.         "url": "https://maps.google.com/?q=O2+CZ"
56.       }
57.     }
58.   }
59. ]
60. }
61. }
62.
```

Display on the OTT will be like below:





#### 4.2.4. Rich Card Carousel

Bot can use the below as a reference to send Rich Card Carousel to user.

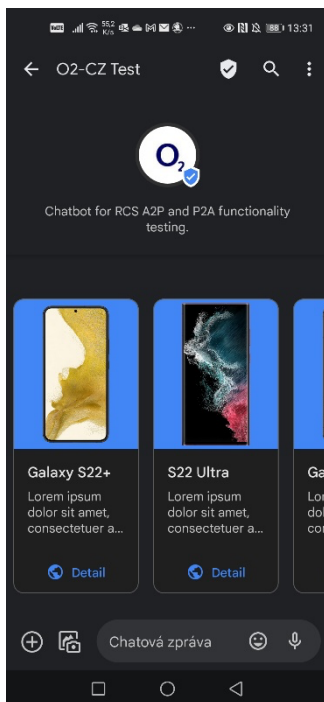
*Rich Card Carousel Request*

```
1. POST https://europe-
rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "richCard": {
8.       "carouselCard": {
9.         "cardWidth": "SMALL",
10.        "cardContents": [
11.          {
12.            "title": "Galaxy S22+",
13.            "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
14.            "media": {
15.              "height": "MEDIUM",
16.              "contentInfo": {
17.                "fileUrl": "https://images.o2.cz/hwapi/2022_9_13/195497CjUHwevs6.png",
18.                "forceRefresh": false
19.              }
20.            },
21.            "suggestions": [
22.              {
23.                "action": {
24.                  "text": "Detail",
25.                  "postbackData": "NOREPLY",
26.                  "openUrlAction": {
27.                    "url": "https://www.o2.cz/telefon-y-a-zarizeni/produkt/samsung-galaxy-s22-plus-5g-128gb-
cerna"
28.                  }
29.                }
30.              }
31.            ]
32.          },
33.          {
34.            "title": "S22 Ultra",
35.            "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
36.            "media": {
37.              "height": "MEDIUM",
38.              "contentInfo": {
39.                "fileUrl": "https://images.o2.cz/hwapi/2022_9_13/195501uxbTiNe43.png",
40.                "forceRefresh": false
41.              }
42.            },
43.            "suggestions": [
44.              {
45.                "action": {
46.                  "text": "Detail",
47.                  "postbackData": "NOREPLY",
48.                  "openUrlAction": {
49.                    "url": "https://www.o2.cz/telefon-y-a-zarizeni/produkt/samsung-galaxy-s22-ultra-5g-128gb-
vinova"
50.                  }
51.                }
52.              }
53.            ]
54.          },
55.          {
56.            "title": "Galaxy S22+",
57.            "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit.",
58.            "media": {
```



```
59.         "height": "MEDIUM",
60.         "contentInfo": {
61.             "fileUrl": "https://images.o2.cz/hwapi/2022_9_13/195497CjUHwevs6.png",
62.             "forceRefresh": false
63.         }
64.     },
65.     "suggestions": [
66.         {
67.             "action": {
68.                 "text": "Detail",
69.                 "postbackData": "NOREPLY",
70.                 "openUrlAction": {
71.                     "url": "https://www.o2.cz/telefon-y-a-zarizeni/produkt/samsung-galaxy-s22-plus-5g-128gb-
cerna"
72.                 }
73.             }
74.         }
75.     ]
76. }
77. ]
78. }
79. }
80. }
81. }
82. }
```

Display on the OTT will be like below:





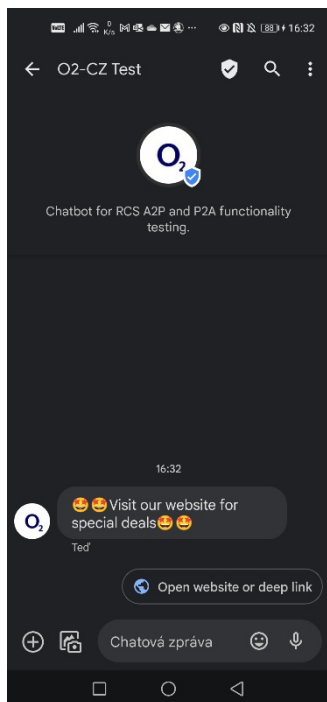
### 4.2.5. URL Action

Bot can use the below as a reference to send Url Action to user.

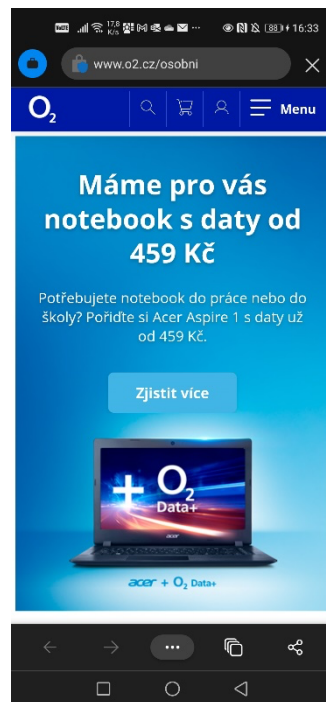
#### URL Action Request

```
1. POST https://europe-rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "text": "👉👉 Visit our website for special deals👉👉",
8.     "suggestions": [
9.       {
10.        "action": {
11.          "text": "Open website or deep link",
12.          "postbackData": "set_by_chatbot_open_url",
13.          "openUrlAction": {
14.            "url": "https://www.o2.cz"
15.          }
16.        }
17.      }
18.    ]
19.  }
20. }
```

Display on the OTT will be like below:



On Clicking the action display will be like below:





### 4.2.6. Map Action

Bot can use the below as a reference to send Map Action to request users location

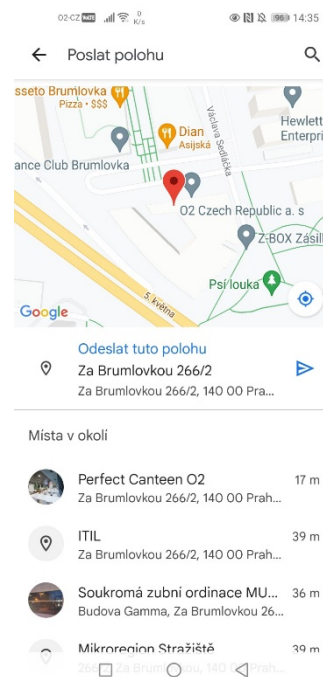
Map Action Request

```
1. POST https://europe-rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "text": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. In enim a arcu imperdiet malesuada.",
8.     "suggestions": [
9.       {
10.        "action": {
11.          "text": "Request a geo location",
12.          "postbackData": "CMD_Command1",
13.          "shareLocationAction": {}
14.        }
15.      }
16.    ]
17.  }
18. }
```

Display on the OTT will be like below:



On Clicking the action display will be like below:





#### 4.2.7. Calendar Action

Bot can use the below as a reference to send Calendar Action to user  
*Calendar Action Request*

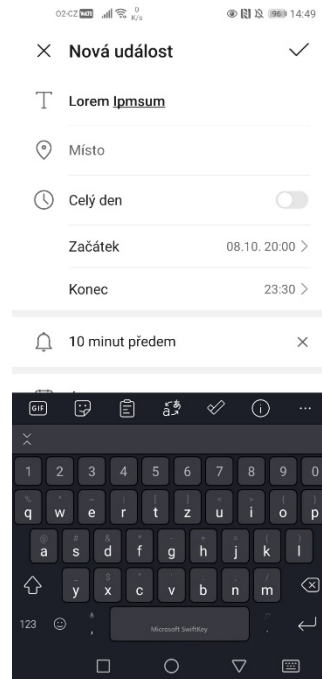
```
1. POST https://europe-
rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "richCard": {
8.       "standaloneCard": {
9.         "cardContent": {
10.          "title": "Lorem ipsum dolor sit amet",
11.          "description": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. In
enim a arcu imperdiet malesuada. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariat.",
12.          "media": {
13.            "height": "TALL",
14.            "contentInfo": {
15.              "fileUrl": "https://www.o2.cz/_pub/a6/fb/2b/237540_504860_02_RGB.jpg",
16.              "forceRefresh": false
17.            }
18.          },
19.          "suggestions": [
20.            {
21.              "action": {
22.                "text": "Add to calendar",
23.                "postbackData": "NOREPLY",
24.                "createCalendarEventAction": {
25.                  "startTime": "2022-10-08T18:00:00Z",
26.                  "endTime": "2022-10-08T21:30:00Z",
27.                  "title": "Lorem Ipsum",
28.                  "description": "Lorem Ipsum"
29.                }
30.              }
31.            }
32.          ]
33.        },
34.        "cardOrientation": "VERTICAL"
35.      }
36.    }
37.  }
38. }
```



Display on the OTT will be like below:



On Clicking the action display will be like below:



#### 4.2.8. Dialer Action

Bot can use the below as a reference to send Dialer Action to user

*Dialer Action Request*

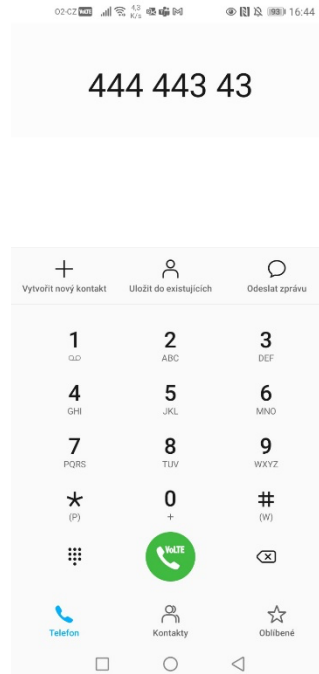
```
1. POST https://europe-rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "text": "☎ Call us...",
8.     "suggestions": [
9.       {
10.        "action": {
11.          "text": "Infolinka",
12.          "postbackData": "NOREPLY",
13.          "dialAction": {
14.            "phoneNumber": "+42044444343"
15.          }
16.        }
17.      }
18.    ]
19.  }
20. }
```



Display on the OTT will be like below:



On Clicking the action display will be like below:



#### 4.2.9. Send Location

Bot can use the below as a reference to send Location to user.

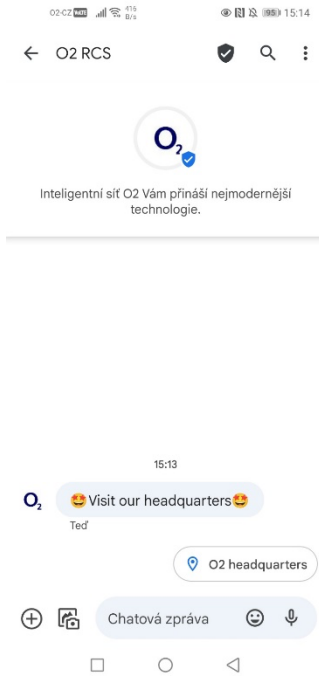
Parameters need to be sent as per RCS Geolocation Push defined in 3.2.6 of RCC.07 Specification.

*Send Location Request*

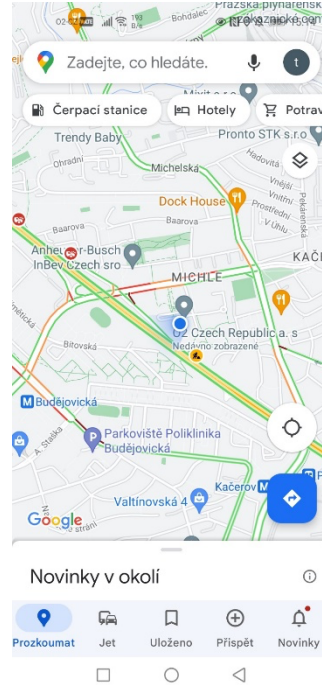
```
1. POST https://europe-
rcsbusinessmessaging.googleapis.com/v1/phones/{{MSISDN}}/agentMessages?messageId={{$guid}}&agentId={{agentId}}
2. Authorization: Bearer {{token}}
3. Content-Type: application/json
4.
5. {
6.   "contentMessage": {
7.     "text": "📍Visit our headquarters📍",
8.     "suggestions": [
9.       {
10.        "action": {
11.          "text": "O2 headquarters",
12.          "postbackData": "NOREPLY",
13.          "viewLocationAction": {
14.            "latLong": {
15.              "latitude": 50.04665461423805,
16.              "longitude": 14.455768573286594
17.            },
18.            "label": "O2 headquarters"
19.          }
20.        }
21.      }
22.    ]
23.  }
24. }
```



Display on the OTT will be like below:



On Clicking the action display will be like below:





### 4.3. User Sending Messaging to Bot

All MO (Mobile Originated) RCS messages from user are sent to you webhook

The "data" field is a base64-encoded string that the agent must decode to match the UserMessage structure.

See section [References](#):

```
1. POST https://your-webhook-url.com/webhook
2. Content-Type: application/json
3. X-Goog-Signature: <HMAC_SHA512_signature>
4. {
5.   "message": {
6.     "data": "<base64_encoded_payload>"
7.   }
8. }
```

#### 4.3.1. Text

*Agent receives text*

```
1. POST https://your-webhook-url.com/webhook
2. Content-Type: application/json
3. {
4.   "agentId": "{{agentId}}",
5.   "senderPhoneNumber": "+420602020202",
6.   "messageId": "msg-123-abc",
7.   "sendTime": "2018-07-13T11:04:20.045123456Z",
8.   "text": "Hi, I need help"
9. }
```

#### 4.3.2. Suggested Response

This is the Suggested Response JSON object following UP 2.0 specification, a.k.a. the "response" object defined in 3.6.10.3 of RCC.07 Specification.

*Agent receives a message from a suggestion*

```
1. POST https://your-webhook-url.com/webhook
2. Content-Type: application/json
3. {
4.   "agentId": "{{agentId}}",
5.   "senderPhoneNumber": "+420602020202",
6.   "messageId": "msg-456-def",
7.   "sendTime": "2018-07-13T11:05:30.045123456Z",
8.   "suggestionResponse": {
9.     "postbackData": "NOREPLY",
10.    "text": "Visit our Web"
11.  }
12. }
```



### 4.3.3. User's Location

*Agent receives a location*

```
1. POST https://your-webhook-url.com/webhook
2. Content-Type: application/json
3. {
4.   "agentId": "{{agentId}}",
5.   "senderPhoneNumber": "+420602020202",
6.   "messageId": "msg-789-ghi",
7.   "sendTime": "2018-07-13T11:06:45.045123456Z",
8.   "location": {
9.     "latitude": 50.0755,
10.    "longitude": 14.4378
11.  }
12. }
13.
```

### 4.3.4. File sharing

*Agent receives a file*

```
1. POST https://your-webhook-url.com/webhook
2. Content-Type: application/json
3. {
4.   "agentId": "{{agentId}}",
5.   "senderPhoneNumber": "+420602020202",
6.   "messageId": "msg-101-jkl",
7.   "sendTime": "2018-07-13T11:07:50.045123456Z",
8.   "userFile": {
9.     "thumbnail": {
10.      "mimeType": "image/jpeg",
11.      "fileSizeBytes": 1280,
12.      "fileUri": "https://storage.googleapis.com/...",
13.      "fileName": "photo.jpg"
14.    },
15.     "payload": {
16.      "mimeType": "image/jpeg",
17.      "fileSizeBytes": 127806,
18.      "fileUri": "https://storage.googleapis.com/...",
19.      "fileName": "photo_original.jpg"
20.    }
21.  }
22. }
23.
```



### 4.3.5. Events (DELIVERED/READ/IS TYPING)

Your agent receives notifications when the messages it sends to users are delivered and read.

```
Possible `eventType`:  
`DELIVERED` - message delivered to user,  
`READ` - message read by user,  
`IS_TYPING` - user is typing
```

```
1. POST https://your-webhook-url.com/webhook  
2. Content-Type: application/json  
3. {  
4.   "agentId": "{{agentId}}",  
5.   "senderPhoneNumber": "+420602020202",  
6.   "messageId": "YOUR_MESSAGE_ID",  
7.   "eventId": "event-123",  
8.   "sendTime": "2018-07-13T11:08:00.045123456Z",  
9.   "eventType": "DELIVERED"  
10. }  
11.
```

## 5. REFERENCES

- [O2 Logo design guidelines for RCS chatbot](#) best practices for Logo and Hero image chatbot's branding
- [GoogleRBM Console - Guides](#) - Develop with RCS Business
- [RBM Administration console](#) - Chatbot agents management.
- [Webhooks](#) – URL where RMB platform post messages and events