

Observações:

- Chave API ou Authorization?
- Formatar como quiser os nomes, valores, tipos dos campos para usar as melhores práticas de DEV.

## **ENVIO DE RCS**

|                 |                                       |
|-----------------|---------------------------------------|
| <b>Endpoint</b> | http://api-plataform.com/v1/rcs/send/ |
| <b>Método</b>   | POST                                  |

### **Entrada**

- headers
  - authorization
  - content-type
- body
  - accountId (int, id da conta)
  - channel (ex: RCS)
  - channelType (text, ex: "Single", "Basic", etc)
  - templateId (text, template a ser usado para disparo - precisa ser um template de acordo com o type)
  - campaignName (opcional, text. Nome do disparo/campanha para agrupar mesmos envios)
  - campaignId (opcional, text. Id do disparo/campanha para agrupar mesmos envios)
  - callbackUrl (opcional, text. url para retorno dos eventos do usuário)
  - fallback (opcional, array. se preenchido e der erro no RCS, dispara SMS)
    - channel (ex: "sms")
    - message ("olá, {{key}}, vamos aproveitar promoção?")
  - messages (array)
    - message
      - number
      - message (opcional se messageTemplateId preenchido)
      - vars:
        - key:value (exemplo: "Nome":"Pedro", "Idade":30)
      - scheduleTo (opcional, horário para disparo da mensagem, se vazio disparo imediato)

## Saída

- “return” (array com retorno da tentativa de acesso a API, definir os nomes dos parâmetros melhor)
  - “return.code” (sucesso a acessar API, erro, etc)
  - “return.message” (mensagem referente ao resultado da chamada da API)
  - “return.campaignName” (nome da campanha enviada pelo cliente)
  - “return.campaignId” (nome da campanha enviada pelo cliente)
  - “return.numberSucesses” (numero de sucessos)
  - “return.numberErrors” (numero de sucessos)
  - “messages” (array com detalhes do disparo)
    - “successes”
      - number
      - callbackMessageId (id que será usado para devolver event no callbackUrl)
    - “errors”
      - number
      - errorMessage

-

## CONSULTA EVENTOS

|                 |  |
|-----------------|--|
| <b>Endpoint</b> | http://api-plataform.com/v1/rcs/events/<br>http://api-plataform.com/v1/rcs/events/:callbackMessageId |
| <b>Método</b>   | GET  |

Obs: mesmo layout de saída quando o callbackUrl é disparado.  
(se passar o callbackMessageId filtra por ele)

## Entrada

- headers
  - authorization
  - content-type
- Body
  - limit (maximo de registros)
  - *page (paginação?)*
  - callbackUserId (opcional, array com ids do usuário)

## Saída

- events
  - eventId
  - callbackMessageId
  - campaignName
  - campaignId
  - templateId
  - templateName
  - accountId
  - channel
  - channelType
  - messageText
  - messageStatus (agendado, enviado, bloqueado, exceção, etc)
  - eventType (click, open, read, reply, etc)
  - eventValue (clicado onde, aberto o que, reposta do usuário, etc)
  - eventDirection (inbound, outbound)
  - callbackUrl
  - scheduleTo
  - createdAt
  - updatedAt
  - timestamp (now())