

**Qualcomm Car-to-Cloud Platform**

**C2D Request API Swagger Usage Document**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** |
| **Name** | Samhitha C S |  |  |
| **Role** |  |  |  |
| **Signature** |  |  |  |
| **Date** | May 18, 2021 |  |  |

May 18, 2021

COGNIZANT

Table of Contents

[1. Swagger UI 2](#_Toc72246252)

[1.1 PushMessageList 3](#_Toc72246253)

[1.1.1 Success Scenario 3](#_Toc72246254)

[1.1.2 Failure Scenario 4](#_Toc72246255)

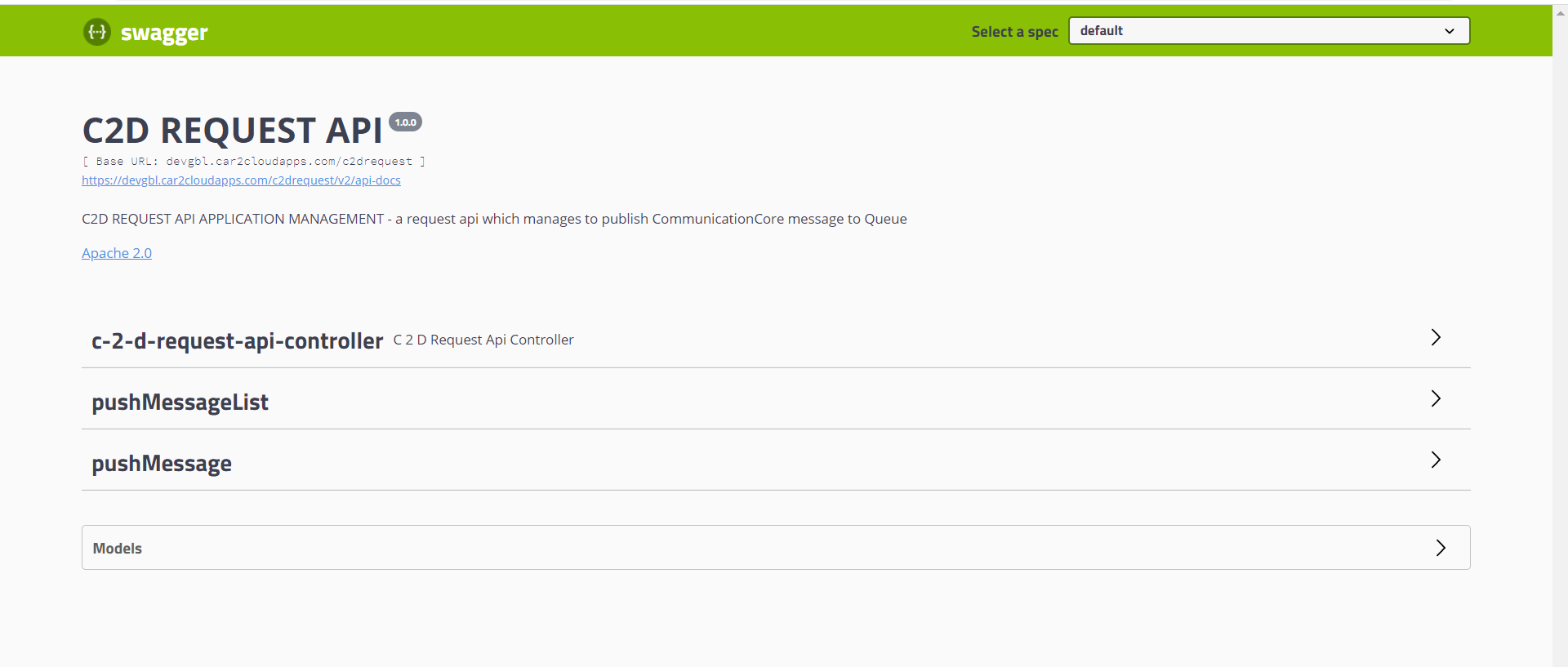
[1.2 PushMessage 6](#_Toc72246256)

[1.2.1 Success Scenario 7](#_Toc72246257)

[1.2.2 Failure Scenario 8](#_Toc72246258)

This document specifies the stepwise details on how to use the Swagger UI for C2D Request API.

### **Swagger UI**



The picture above shows the Swagger UI for C2D Request API .Following are the details:

1. Heading: “C2D Request API”
2. Title: “C2D Request API Application Management”
3. Version: 1.0.0
4. Post Method : “PushMessageList”
5. Post Method : “PushMessage”

The API designed accepts a message request from the User/Application and publishes the request to the inbound queue. Here the Swagger UI helps to post the message request to the API and gets the response.

## **PushMessageList**

The Message request that the API receives is a valid message request. If the user/Application wants to send a list of messages, then they can opt for PushMessageList. There is an option to configure the list size in AWS code commit. If the list size is specified as -1 then there is no limit to the number of messages that can be sent on a message list, any other value specified, then that would act as the list size limit.

There are 4 test scenarios for PushMessageList:

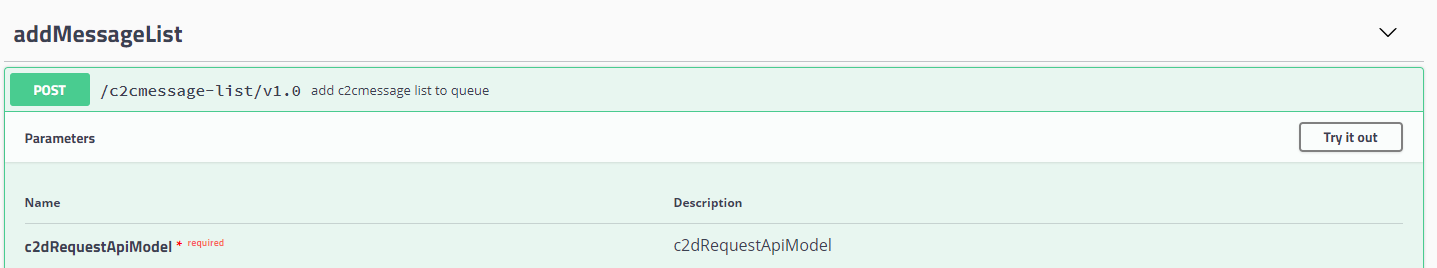
1. Success
2. Error Code 10016: List size limit exceeded.
3. Error Code 10014: Empty Message List.
4. Error Code 10011: Invalid Queue name or Queue not found.
5. Error Code 10013: Publish failed due to some internal error.

The Error Code 10013 cannot be demonstrated as no internal error occurred at the time of execution.

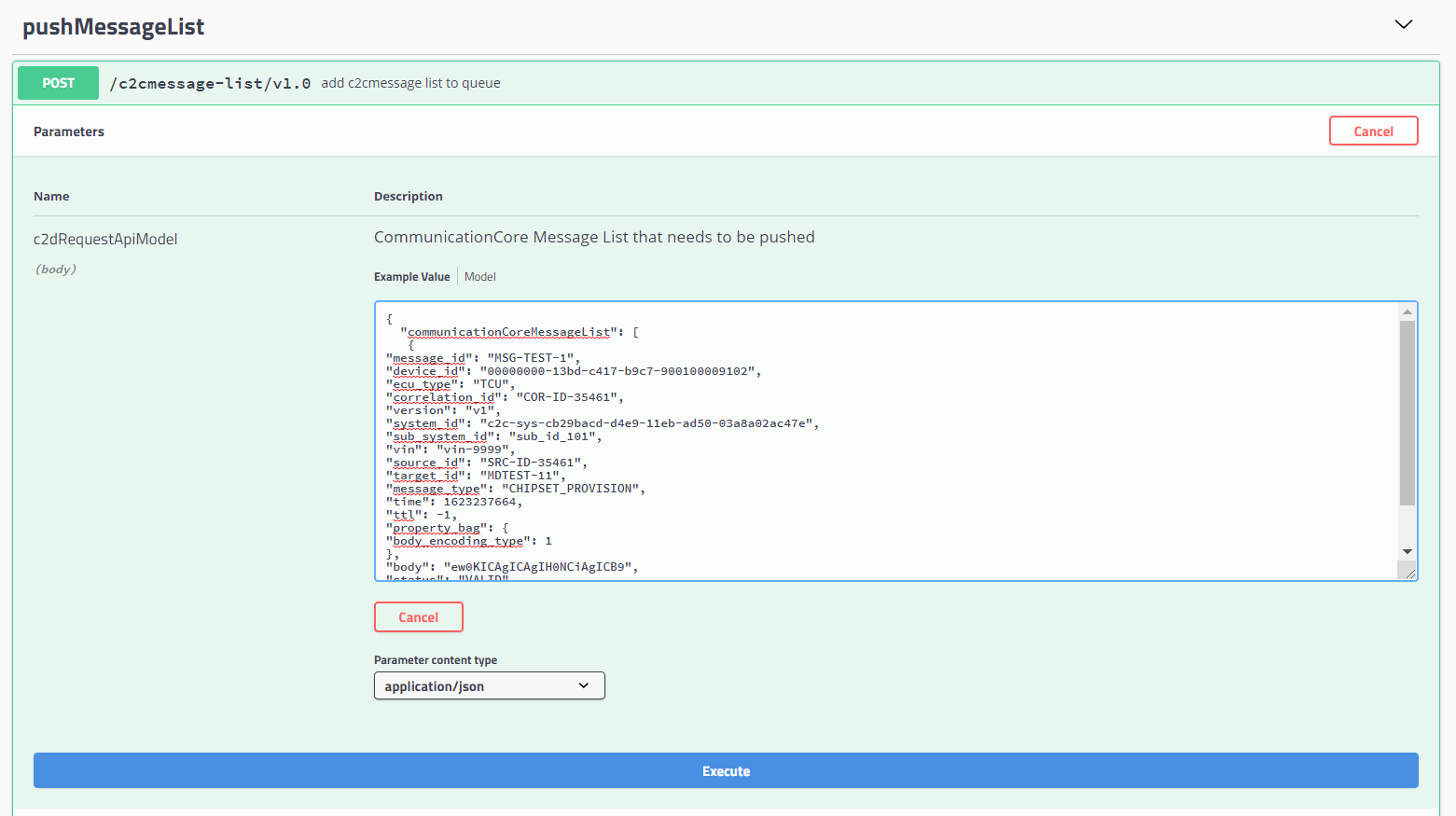
## **1.1.1 Success Scenario**

Steps Followed:

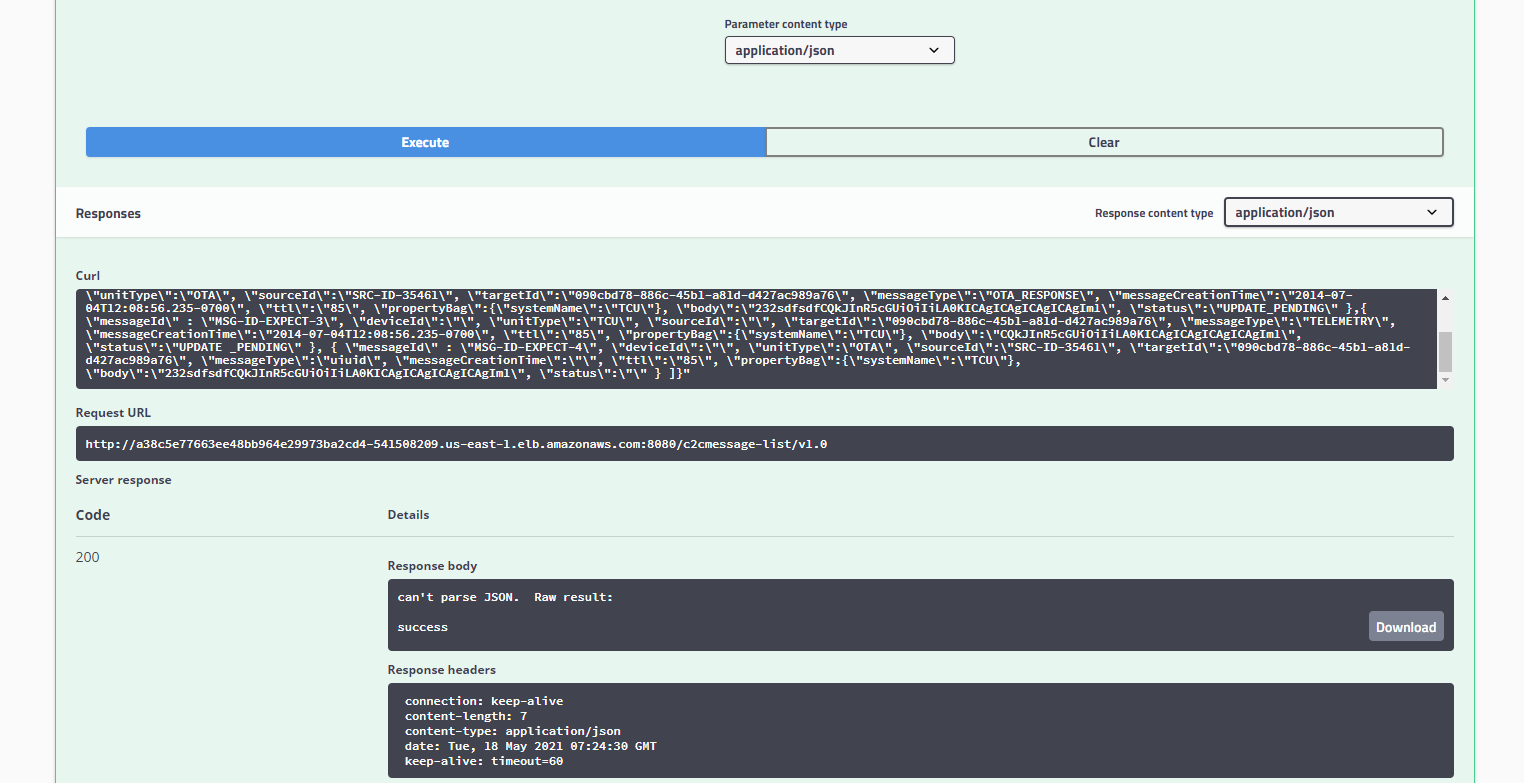
1. Click on the Arrow that is adjacent to PushMessageList
2. Click on the POST option
3. Select the Try it out Button on the right



1. Enter the Message list
2. Click on Execute



1. Check the Response

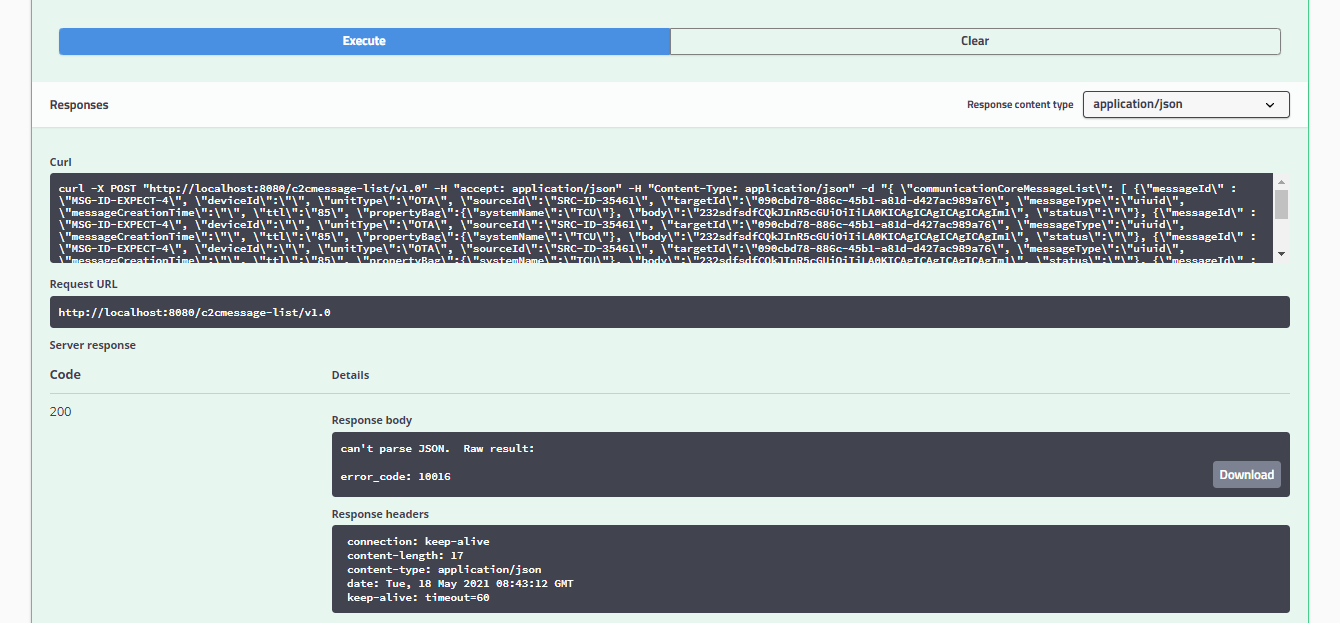


## **1.1.2 Failure Scenario**

* Error Code 10016

Steps Followed:

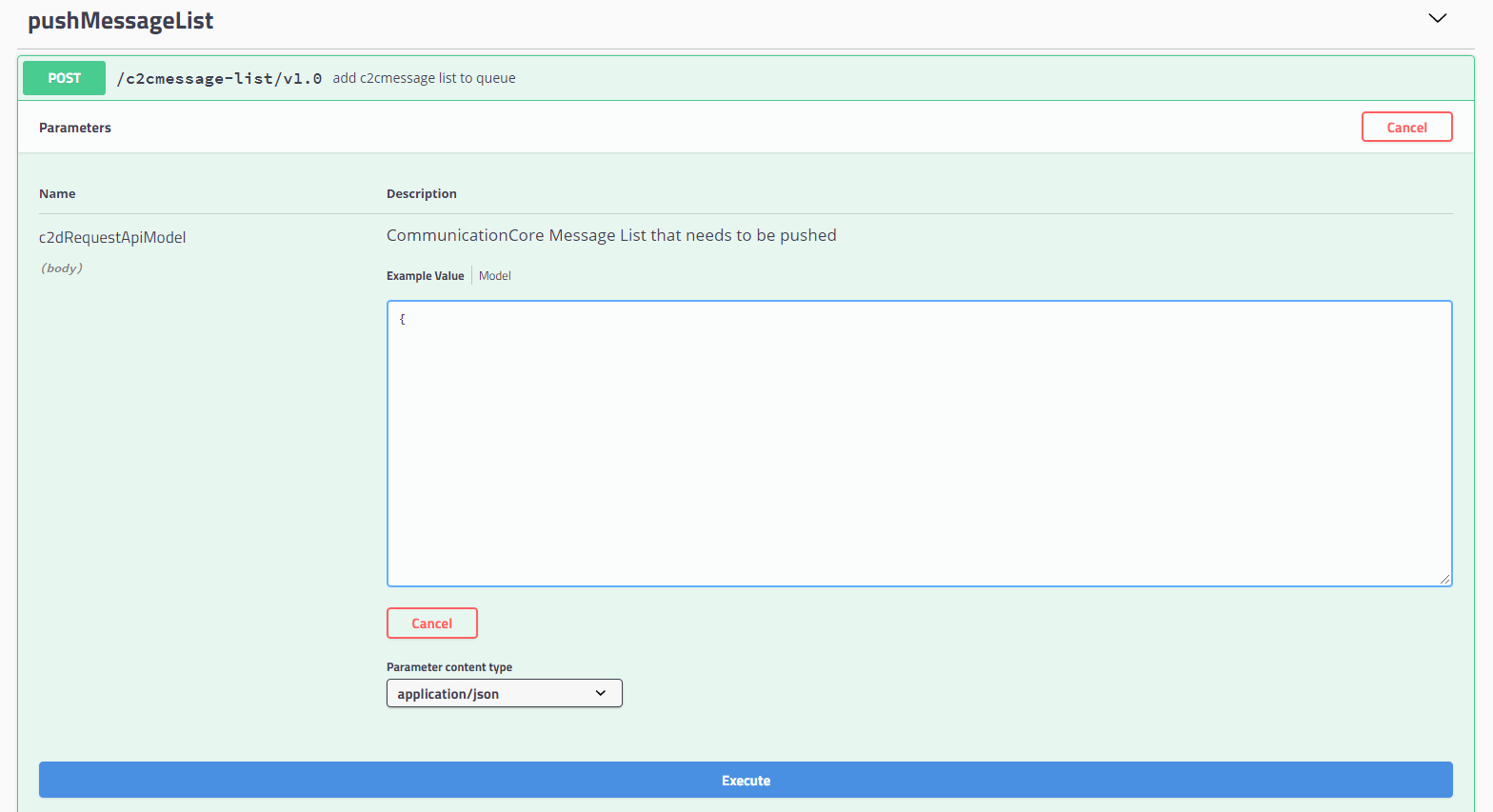
1. Repeat steps 1 to 5 of Success Scenario
2. The list size has exceeded the limit ,hence the response

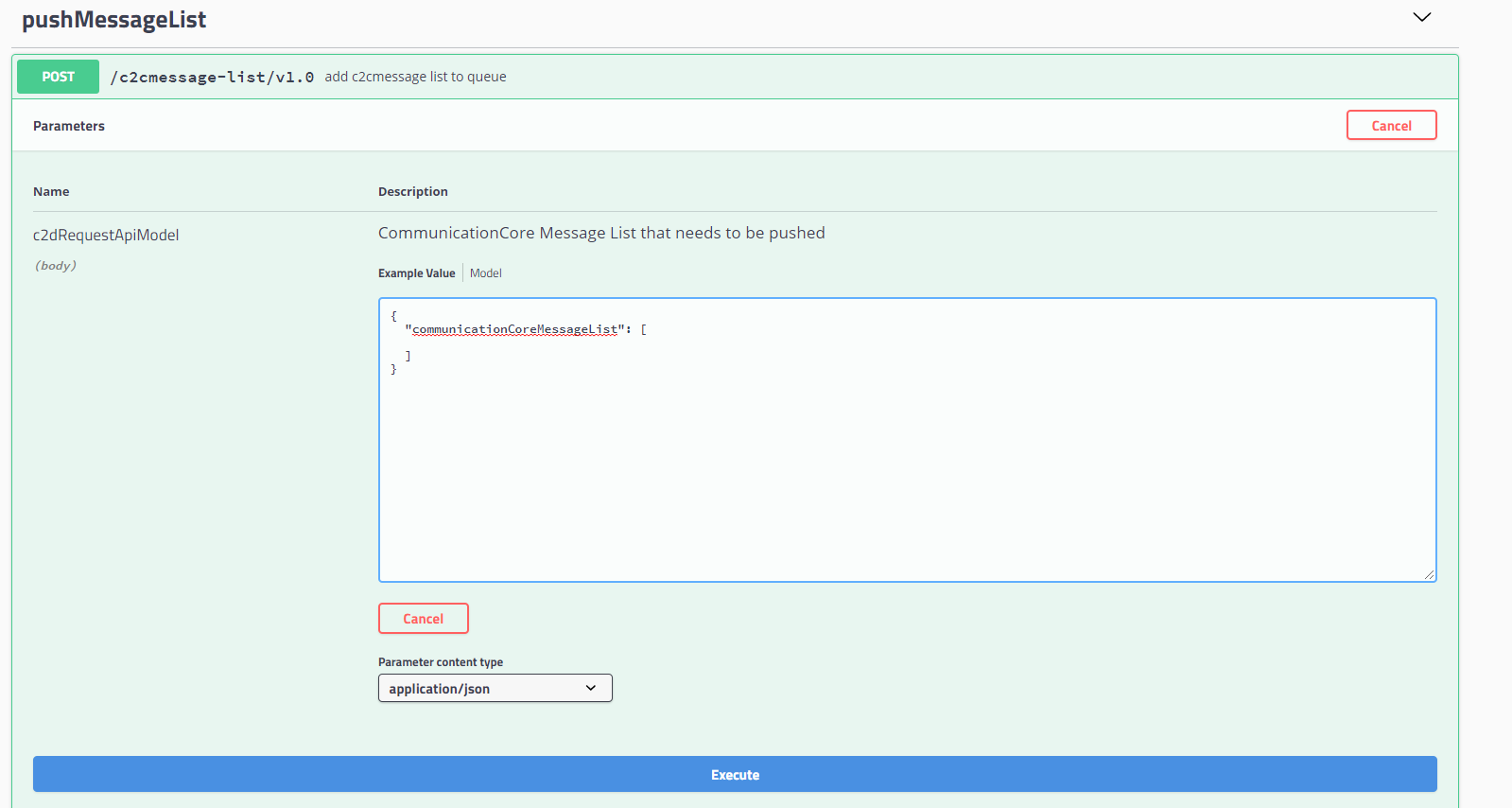


* Error Code 10014

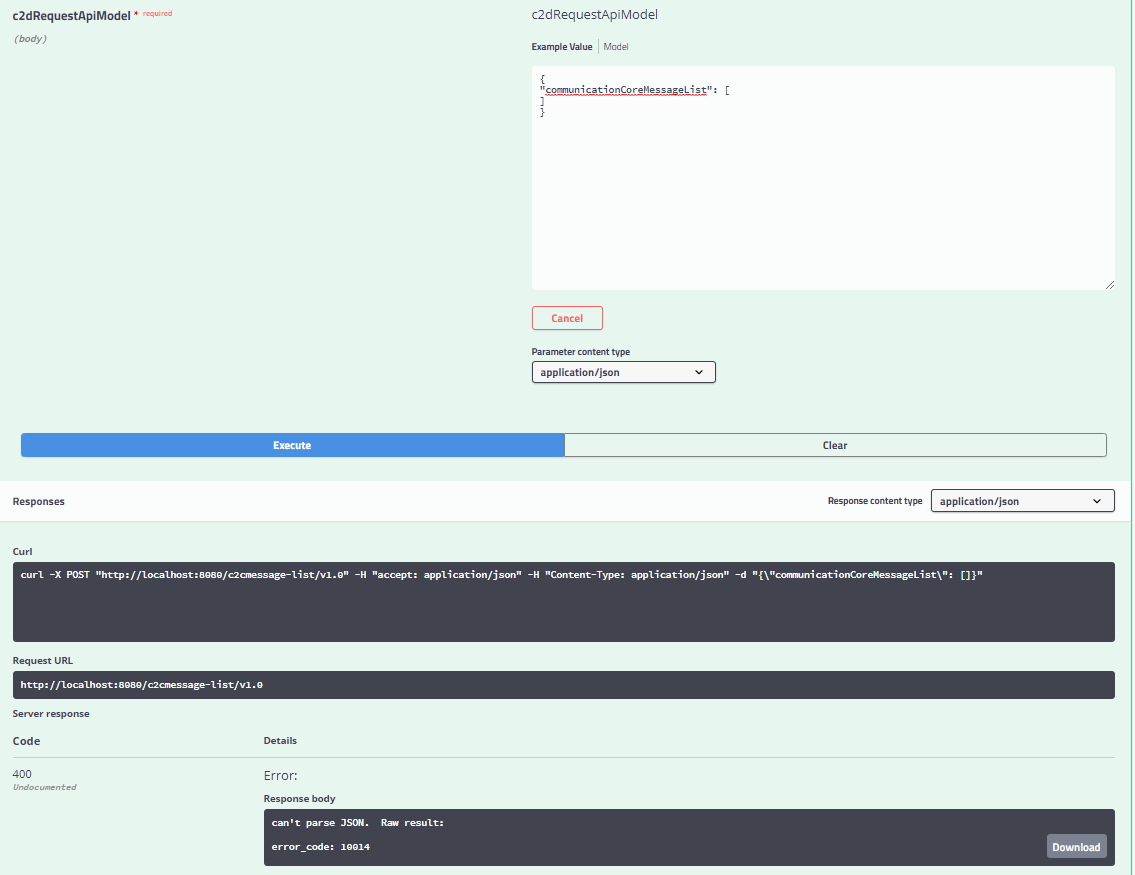
Steps Followed:

1. Repeat steps 1 to 3 of Success Scenario and the message list is of the form {} or { “communicationCoreMessage”:[]}
2. Click on Execute





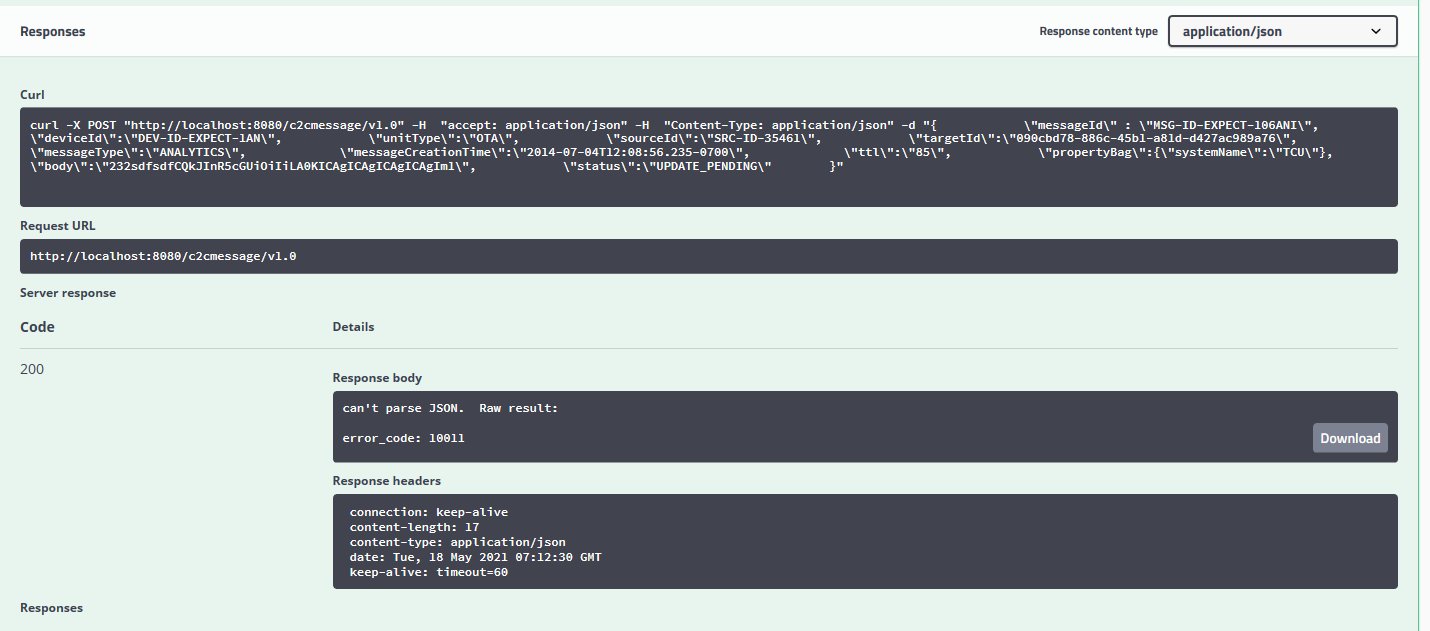
1. As there are no messages in the message list, the Response for both cases



* Error Code 10011

Steps Followed:

1. Repeat steps 1 to 5 of success scenario
2. Due to Incorrect queue name or a Correct queue name in wrong region this error occurs. Hence the response



## **PushMessage**

The Message request that the API receives is a valid message request. If the user/Application wants to send a single message, then they can opt for PushMessage.

There are three Test Scenarios for PushMessage:

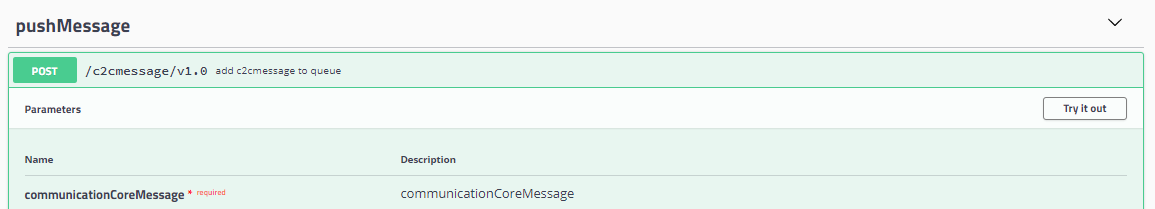
1. Success
2. Error Code 10011: Invalid Queue name or Queue not found.
3. Error Code 10013: Publish failed due to some internal error.

The Error Code 10013 cannot be demonstrated as no internal error occurred at the time of execution.

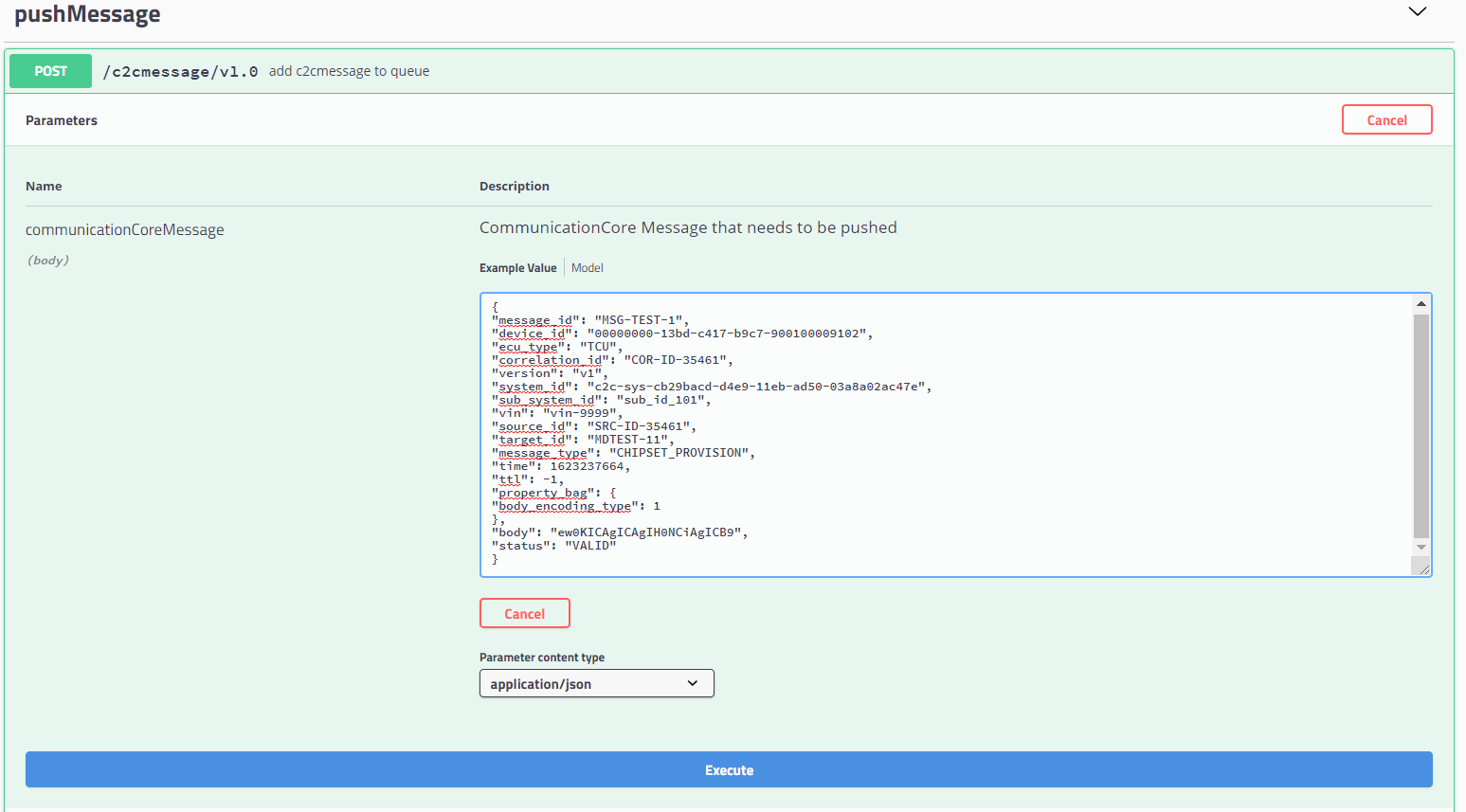
## **Success Scenario**

Steps Followed:

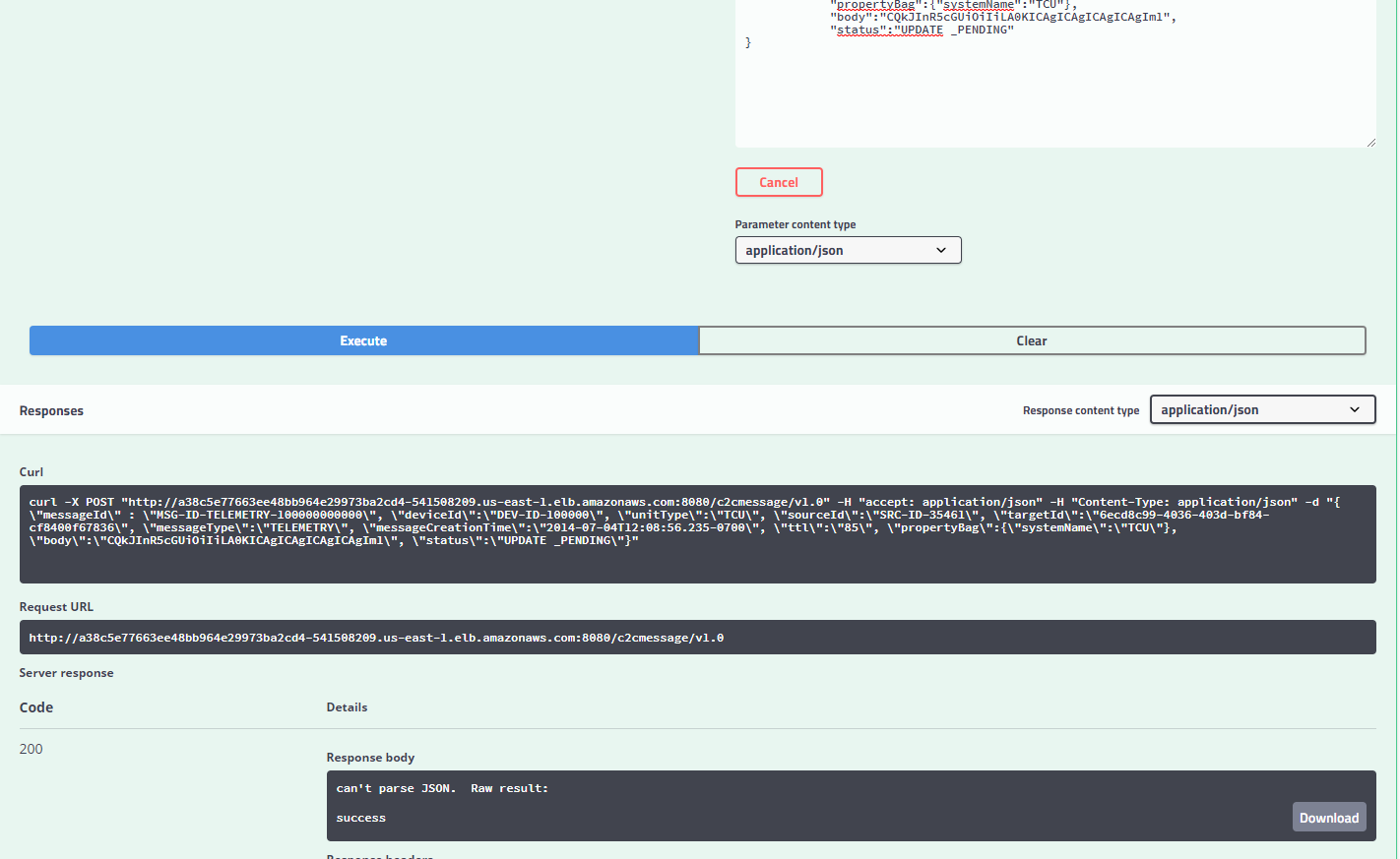
1. Click on the Arrow that is adjacent to PushMessage
2. Click on the POST option
3. Select the Try it out Button on the right



1. Add the Message
2. Click on Execute



1. Check the Response



## Failure Scenario

Error Code 10011

Steps Followed:

1. Repeat steps 1 to 5 of success scenario
2. Due to Incorrect queue name or a correct queue name in wrong region this error occurs. Hence the response

