Critico

Push Service REST API

Push Service REST API | Version 1.6

User Guide

Table of Contents

Amendment History 3

Critico Rest API 4

Overview 4

Push Service Management 4

Push Specification 5

Report 5

Message 5

Responder Message 6

Appendix A - Delivery Receipt Result Codes 8

Appendix B - Responder Message Type 9

Appendix C – Example Java code 10

Amendment History

Date	Update
10/02/2012	Spelling correction of Push Specification resources
25/04/2012	Update CallbackResponse class with new “response” variable
14/01/2013	Update current documentation
10/04/2014	Amend report title – page 5
16/09/2014	Added Responder Application message types to callback
17/09/2015	Added clientTransactionID in delivery report callback
06/09/2019	Amended Push Service Registration

Critico Rest API

Overview

Critico Push Service provides users with the ability to receive messages, delivery reports and responder messages that are sent from Critico's messaging platform. Once the implementation is complete, according to the specification stated in this document, the user can assess and process the information.

In order to utilise this service the user needs to obtain a Critico account and have at least one push receiver registered with Critico.

Push Service Registration

In order to utilise the Push Service, you need to register your implementation/receiver with Critico, against an active Critico account.

To register your Push Service end-point you should provide the following information to our Customer Support Team at customersupport@criticogroup.com

1. The Critico account username against which you wish the push service to operate
2. Your end-point (the URL address and port to which you wish Critico to push).
3. The Type of message information you wish to receive (Delivery Receipts, Reply messages, Responder pager messages).

Please ensure that your Firewall ACL allows traffic from the below IP ranges:

91.220.160.64/28
91.220.160.192/28

Ports that can be used for the Push Service are 80, 8080, 443 & 9110 - 9119.

Push specification

Clients can implement a rest based push receiver according to the following specification.

POST Report /deliveryreport/receive

This resource will receive delivery notifications of sent messages.

Request Parameters

Field Name	Definition
destination	The recipient's address
resultCode	The status code (Appendix A)
receiptTime	The status update time
source	Source address
transanctionID	Unique ID assigned to the transaction
clientTransactionID	Reserved: Will pass back the "client assigned id" of the original message if one had been specified

Response

Output	Definition

Application/json	{
	"@id": "200",
	"#text": "OK"
	}

Application/xml	<response id="200">OK</response>

POST Message /message/receive

This resource will receive notifications of received messages

Request Parameters

Field Name	Definition
destination	The recipient's address
messageTime	The time the message was received
text	Message text
source	Source address
transanctionID	Unique ID assigned to the transaction

Response

Output	Definition

Application/json	{
	"@id": "200",
	"#text": "OK"
	}

Application/xml	<response id="200">OK</response>

POST	Responder Message /respondermessage/receive

This resource will receive notifications of sent responder messages.

Request Parameters

Field Name	Definition
battery	The battery level as percentage
delivered	Message delivered **Value** **Description** 0 Not delivered 1 Delivered
read	Specifies if message was read **Value** **Description** 0 Not read 1 Read
destination	Destination device address
source	Source device address
locationLatitude	The latitude of device
locationLongitude	The longitude of device
locationTimestamp	Time that the position was generated
locationAccuracy	GPS location accuracy (not present for type="Pager")
text	Message content
originalText ***{po}***	Original message text when replying
fixedText ***{po}***	Customer specific status mode of device e.g. available
fixedTextID ***{po}***	An ID for status mode [2-9]
timestamp	Time the message was received
txID	Unique transaction ID for the message
messageType	Type of message (Appendix B)
deviceType	Device operating system
uID	Internal Critico unique id

Response

Output	Definition

Application/json	{
	"@description":"DataMessage Processed",
	"@id": "200",
	"#text": "OK"
	}

Application/xml	<response id="200">OK</response>

{po} *: only applicable for type="Pager"*

Responder App options by message type

Field Name	Acknowledgment	Delivery Receipt	Read Receipt	Lone Worker - Started - Updated - Stopped SOS
battery	✓	✓	✗	✓
delivered	✗	✓	✓	✗
read	✗	✓	✓	✗
destination	✓	✓	✓	✓
source	✓	✓	✓	✓
locationLatitude	Optional	Optional	✗	Optional
locationLongitude	Optional	Optional	✗	Optional
locationTimestamp	Optional	Optional	✗	Optional
locationAccuracy	Optional	Optional	✗	Optional
text	✗	✓	✓	✗
originalText	✓	✓	✓	✓
timestamp	✓	✓	✓	✓
txID	✓	✓	✓	✓
messageType	180	182	200	160/162/161/32
deviceType	✓	✓	✓	✓

Appendix A A Delivery Receipt's Result Codes

These are message statuses that are generated after the Critico gateway has accepted the message for delivery.

Code	Description	In Detail
200	Sent	Message has been delivered to handset
201	Accepted	Message has been received and is being sent to handset
202	Group Accepted	Group message has been received and sent to the group device number
214	TTS Accepted	Message has been accepted as type TTS
203	Valid Login	Login Successful
400	No Access to this Service	You have not signed up to the service you are trying to use
401	Message Not Sent	Rejected by Remote Provider
402	Service currently unavailable	Service connection is currently unavailable
403	Could not send e-mail	Unable to convert message / device into e-mail
404	Unable to deliver	Unspecified protocol error on the MT handset
501	Bad Character	Content related error
502	Invalid Number	The device number you are sending to is incorrect
503	Unknown Called Address	The device number you are sending to does not exist
504	Unknown Pager	The device number you are sending to does not exist
505	Protocol Error	Permanent operator error
507	Insufficient Credits	Your account does not presently possess enough message credits to send your request
551	Failed Security	MT handset is listed as an Illegal device on the MSC.
552	Inactive Subscriber	The device number is inactive
553	Max Limit Reached	The set amount of messages for this service has been reached (Generally applies to our trial customers)
554	Please Contact Service Provider	Operator network failure
555	Systems Error	Service connection is currently unavailable
556	Unknown	Final status is unknown
557	Expired by Provider	Unable to deliver within the allocated time frame
558	Failed to replace	MT network unable to replace the SMS on the MT customer's handset
559	Insufficient Funds	Credit related - message has not been retried by operator

Appendix B A Responder Message Type

Type	Description
1	Auto acknowledgement
2-9	Fixed text response
17	Location request response
32	SOS
35	SOS test
38	SOS update (Later version 2.43 onwards)
48	Status switch
64	Remote tracking
100	Lone worker
110	Free text response

Appendix C – Example Java code

Type	**Example Java code**
Message	@POST @Path(*"/message/receive"*) @Produces({MediaType.APPLICATION_XML,MediaType.APPLICATION_JSON}) **public** Object **receiveMessage** (@FormParam(*"destination"*) String destination, @FormParam(*"messageTime"*) String messageTime, @FormParam(*"source"*) String source, @FormParam(*"text"*) String text, @FormParam(*"transactionID"*) String transactionID)
Delivery Report	@POST @Path(*"/deliveryreport/receive"*) @Produces({MediaType.APPLICATION_XML,MediaType.APPLICATION_JSON}) **public** Object **receiveDeliveryReport** (@FormParam(*"destination"*) String destination, @FormParam(*"resultCode"*) String resultCode, @FormParam(*"receiptTime"*) String receiptTime, @FormParam(*"source"*) String source, @FormParam(*"transactionID"*) String transactionID)
Responder Message	@POST @Path(*"/respondermessage/receive"*) @Produces({MediaType.APPLICATION_XML,MediaType.APPLICATION_JSON}) **public** Object **receiveResponderMessage** (@FormParam(*"battery"*) String battery, @FormParam(*"delivered"*) String delivered, @FormParam(*"read"*) String read, @FormParam(*"destinationFN"*) String destinationFN, @FormParam(*"destinationSN"*) String destinationSN, @FormParam(*"destination"*) String destination, @FormParam(*"sourceFN"*) String sourceFN, @FormParam(*"sourceSN"*) String sourceSN, @FormParam(*"source"*) String source, @FormParam(*"locationLatitude"*) String locationLatitude, @FormParam(*"locationLongitude"*) String locationLongitude, @FormParam(*"locationTimestamp"*) String locationTimestamp, @FormParam(*"locationAccuracy"*) String locationAccuracy, @FormParam(*"originalText"*) String originalText, @FormParam(*"text"*) String text, @FormParam(*"fixedText"*) String fixedText, @FormParam(*"fixedTextID"*) String fixedTextID, @FormParam(*"timestamp"*) String timestamp, @FormParam(*"txID"*) String txID, @FormParam(*"messageType"*) String messageType, @FormParam(*"deviceType"*) String *deviceType*, @FormParam(*"uID"*) String uID)