Colt TMF 621 eBonding Ticketing API Specification

SIGNATURES CONTOL

Version	Prepared by	Reviewed by	Approved by
1.0	Shyam Kumar	Spandan Koti	Andrew Gummer & Viswanathan Kandasamy

Modifications Control

Version	Date	Author	Reason For Change
V1.0	08.02.2022	Shyam Kumar S	Final Version

ÍNDEX

1. GLOSSARY OF TERMS....4
2. INTRODUCTION....4
2.1 Document Purpose....4
3. TECHNICAL INFORMATION....4
3.1 Security....4
3.2 Base URL....5
4. BUSINESS PROCESSES....5
4.1 Reactive Ticket Process....5
4.2 Proactive Ticket Process....6
4.3 Cancel Ticket....7
5 INTEGRATION SCENARIOS....8
5.1 Customer Triggered Transactions....8
5.1.1 Create Ticket....9
Please Refer the below Section "**Synch Response / Acknowledgement"**....11
5.1.2 Update Ticket....11
5.1.3 Verify Ticket Resolution....13
5.1.4 Cancel Ticket Resolution....15
Synch Response / Acknowledgement....16
5.2 COLT Triggered Transactions....17
5.2.1 Proactive Ticket Notification....18
5.2.2 Colt Comment Update....19
5.2.3 Colt Sets Stop Clock for an Incident....19
5.2.4 Colt Removes Stop Clock for an Incident....19
5.2.5 Colt Cancels Incident....20
5.2.6 Colt Resolves Incident....20
5.2.7 Colt Closes Incident....20
5.3 Notify Output Data Sample....20
5.4 Notify Ticket Sample....21
6 PRIORITY AND STATUS VALUES....21
6.1 Impact, Urgency & Priority Matrix....21
6.2 Status Values....21
7 MAPPING SHEET....21
8 TMF621 TROUBLE TICKET API SPECIFICATION....22
9 SWAGGER FILES....22

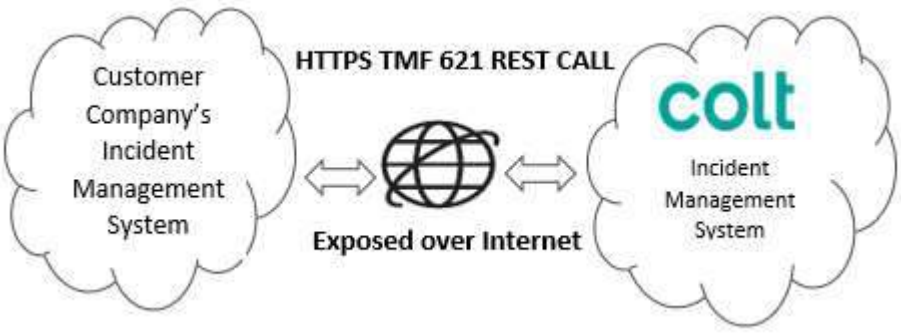
1. GLOSSARY OF TERMS

Term	Description
API	Application Programming Interface
POST	HTTP method of an API used to create a resource
PATCH	HTTP method of an API used to perform partial update of a resource
TMF	TM Forum

2. INTRODUCTION

At present, Colt is having a eBonding Ticket management process which is unique to Colt. Colt is adapting TMF Open API specification to meet the industry standards and to generalise the eBonding Ticket management process and related communication with customers. The "TMF621 Open API - eBonding Ticket Management API" provides REST API specification for eBonding Ticket management.

Colt eBonding is a secure, bi-directional system-to-system interface, allowing the customer to connect their internal systems to the Colt Incident Management Systems. This will enable the customer to manage its network easily and provides them with the ability to electronically report and track shared incidents with Colt.
Colt eBonding enables the customer to electronically manage their incidents with Colt using a web service interface. The interface uses standardised REST JSON messages to facilitate a number of Incident Management operations.



2.1 Document Purpose

This document details the strategy and process to adapt the TMF Open API specification for eBonding Ticket management (TMF 621). The document explains the current eBonding Ticket management at Colt and extending TMF change management API for Colt's requirement.

3. TECHNICAL INFORMATION

3.1 Security

TMF 621 REST operations will have to follow basic user authentication process. HTTPS must be used in all the interactions.

What is Required:

- An SSL secured connection between your company's trouble incident system and Colt eBonding Ticketing Interface via an SSL protected internet gateway
- To meet Colt corporate standards, Colt uses port 443 with SSL encryption for all communications across the Internet. Colt requires one-way SSL to authenticate to Colt web service due to internal Colt firewall IP filtering. The Customer will trust Colt Certificate Authority (CA) as a Trusted CA in their application.
- Colt can provide certificate, or Customer can pull certificate chain.

3.2 Base URL

Environment	URL
Prod	https://wmisd012.colt.net/troubleTicketManagement/v1/troubleTicket?jsonFormat=stream
Pre-Prod	https://wmisd112.colt.net/troubleTicketManagement/v1/troubleTicket?jsonFormat=stream
UAT	https://wmisd212.colt.net/troubleTicketManagement/v1/troubleTicket?jsonFormat=stream

4. BUSINESS PROCESSES

4.1 Reactive Ticket Process

Source	Target	Integration Scenario Name	System Interaction Scenarios	Message Type	Transactions Type
Customer	Colt	**Create Ticket**	Create Ticket Request	POST - Trouble Ticket	Inbound
Customer	Colt		Create Ticket Async Response	Trouble Ticket Create Event	Outbound
Customer	Colt	**Update Ticket**	Update Ticket Request	PATCH - Trouble Ticket	Inbound
Customer	Colt		Update Ticket Async Response	Trouble Ticket Attribute Value Change Event	Outbound
Customer	Colt	**Verify Ticket Resolution**	Verify Ticket Resolution Request	PATCH - Trouble Ticket	Inbound
Customer	Colt		Verify Ticket Async Resolution Response	Trouble Ticket Attribute Value Change Event	Outbound

Reactive Ticket Process Step Description

1) A new ticket is created in the customer Incident management system (IMS) which triggers a ticket request to be sent across the eBonding interface to Colt. The ticket request is received by Colt, validated and Colt sends synchronous Acknowledgement in the first level. Once it is

processed further in Siebel a second level Asynch Response along with Ticket ID is sent to the customer.

2) A ticket update is made to the customer ticket and sent to Colt. This could be, for example, a comment or escalation on a ticket, The ticket request is received by Colt, validated and Colt sends an automated Ticket request Acknowledgement in the first level. Once it is processed further in Siebel a second level Asynch Response along with the updated status is sent to the customer.

3) Colt sets a clock stop on their ticket. The Colt ticket 'Status' is moved to "Pending".

4) Colt removes a clock stop from their ticket. The Colt ticket 'Status' is moved back to the 'Status' it was before the clock stop was set, this could be either "Assigned" or "In Progress".

5) Colt resolves their ticket.

6) The customer rejects the ticket resolution. When this rejection is received the Colt ticket 'Status' is moved out of "Resolved". The ticket request is received by Colt, validated and Colt sends an automated Ticket request Acknowledgement in the first level. Once it is processed further in Siebel a second level Asynch Response along with Ticket rejection status is sent to the customer.

7) Colt resolves their ticket after further investigation

8) The customer accepts the ticket resolution. The ticket request is received by Colt, validated Colt sends an automated Ticket request Acknowledgement in the first level. Once it is processed further in Siebel a second level Asynch Response along with Ticket Resolutions is sent to the customer.

9) The Colt ticket is closed.

4.2 Proactive Ticket Process

Source	Target	Integration Scenario Name	System Interaction Scenarios	Message Type	Transactions Type
Colt	customer	**Proactive Ticket Notification**	This transaction shall be used by Colt to send a proactive ticket request to the customer.	Trouble Ticket Create Event	Outbound
Colt	customer	**Colt Comment Update**	This transaction shall be used by Colt to send comments to the customer.	Trouble Ticket Attribute Value Change Event	Outbound
Colt	customer	**Colt Sets Clock Stop for an Incident**	This transaction shall be used by Colt to send the Clock Stop Update.	Trouble Ticket Status Change Event	Outbound
Colt	customer	**Colt Removes Clock Stop for an Incident**	This transaction shall be used by Colt to send the removal Clock Stop Update.	Trouble Ticket Status Change Event	Outbound
Colt	customer	**Colt Resolves Incident**	This transaction shall be used by Colt to send ticket resolution details to the customer.	Trouble Ticket Resolved Event	Outbound

Colt	customer	**Colt Closes Incident**	This transaction shall be used by Colt to send ticket closure details to the customer.	Trouble Ticket Status Change Event	Outbound

Proactive Ticket Process Step Description

1) A new ticket is initiated by the Colt Incident management system (IMS) which sends an Asynchronous TroubleTicketCreateEvent notification to be sent across to the customer. The ticket is received by the customer and a Confirmation Notification Receipt is initiated by the customer back to the Colt.

2) A ticket update is made to the customer ticket and Colt updates the ticket and it triggers the notification event to customer. This could be, for example, a comment or a change in ticket 'Priority'. Colt Sends the ticket update details to the customer via "TroubleTicketAttributeValueChangeEvent" notification.

3) Colt sets a clock stop on their ticket. The Colt ticket 'Status' is moved to "Pending" where a TroubleTicketStatusChangeEvent will be triggered to the customer with Status as Pending.

4) Colt removes a clock stop from their ticket. The Colt ticket 'Status' is moved back to the 'Status' it was before the clock stop was set, this could be either "Assigned" or "In Progress", where a TroubleTicketStatusChangeEvent will be triggered to the customer with Status as In Progress.

5) Colt resolves their ticket & it triggers TroubleTicketResolvedEvent notification to the customer.

6) The customer rejects the ticket resolution. When this rejection is received the Colt ticket 'Status' is moved out of "Resolved". The ticket request is received by Colt, validated and Colt sends an automated Ticket request Acknowledgement in the first level. Once it is processed further in Siebel a second level Asynch Response.

7) Colt resolves their ticket after further investigation & triggers TroubleTicketResolvedEvent to the customer.

8) The customer accepts the ticket resolution. The ticket request is received by Colt, validated and Colt sends an automated Ticket request Acknowledgement in the first level. Once it is processed further in Siebel a second level Asynch Response.

9) The Colt ticket is closed, where an TroubleTicketStatusChangeEvent will be triggered to the customer with Status as closed.

4.3 Cancel Ticket

Source	Target	Integration Scenario Name	System Interaction Scenarios	Message Type	Transactions Type
Customer	Colt	**Cancel Ticket**	Cancel Ticket Request	PATCH - Trouble Ticket	Inbound

Customer	Colt		Cancel Ticket Async Response	Trouble Ticket Attribute Value Change Event	Outbound
Colt	Customer	**Colt Cancels Incident**	This transaction shall be used by Colt to send proactive ticket cancellation details to the customer.	Trouble Ticket Status Change Event	Outbound

1) A reactive ticket is cancelled in the customer Incident management system (IMS) which triggers a cancellation request to be sent across the eBonding interface to Colt. The ticket request is received by Colt, validated and Colt sends an automated Ticket request Acknowledgement in the first level. Once it is processed further in Siebel a second level Asynch Response along with Ticket cancellation status is sent to the customer & no further notification event is sent to customer after ticket cancellation.

2) A proactive ticket is cancelled in the Colt Incident management system (IMS) which triggers a notification event to indicate status change to be sent across the eBonding interface to the customer. The cancellation request is received by the customer, validated and the ticket is cancelled in the customer IMS.

5 INTEGRATION SCENARIOS

5.1 Customer Triggered Transactions

Integration Scenario Name	System Interaction Scenarios	Message Type(Elements)	Transactions Type
Create Ticket	Create Ticket Request	**CreateTicket**	Inbound
	Create Ticket Response	**Ticket Creation Response**	Outbound
Update Ticket	Update Ticket Request	**ModifyTicket**	Inbound
	Update Ticket Response	**Update Ticket Response**	Outbound
Verify Ticket Resolution	Verify Ticket Resolution Request	**VerifyTicketResolution**	Inbound
	Verify Ticket Resolution Response	**Verify Ticket Resolution Response**	Outbound
Cancel Ticket	Cancel Ticket Request	**CancelTicket**	Inbound

	Cancel Ticket Response	**Cancel Ticket Response**	Outbound

5.1.1 Create Ticket

Integration Scenario Name	**Create Ticket**
Interface	API Gateway
Recipients	Vendors
Transport	REST over HTTPS
Operation Name	POST
Description	Create a new TMF 621 ticket in Siebel. It is an asynchronous operation, this means that some time passes between the moment the request is sent and the time the ticket is created. It is not possible to create more than one Ticket with the same CI without closing or cancelling the open Ticket.
Input Data (parameters)	

Name	Type	Mandatory	Description	Value
description	String	Yes	Ticket description	
externalId	String	Yes	External system reference	
name	String	Yes	Name of the ticket or short description	
requestedResolutionDate	String	Yes	Subject that the ticket is being created on. Service/NEID in on Product tab of ticket (like Customer Circuit ID)	
severity	String	Yes	>> severity of "**1**" Colt will set 'Urgency' to "1-Critical", set 'Impact' to "Extensive/Widespread" and Priority as "Urgent" >> severity of "2" Colt will set 'Urgency' to "2-High", set ' Impact' to "Significant/Large" and Priority as "High" >> severity of "3" Colt will set 'Urgency' to "3-Medium", set ' Impact' to "Moderate/Limited" and Priority as "Medium" >> severity of "4" Colt will set 'Urgency' to " 4-Low ", set ' Impact' to "Minor/Localized" and Priority as "Low"	

ticketType	String	Yes	>> ticketType of "**103**" (CircuitDead), Colt will set 'issueSummary' to "Customer Operations / Circuit Down" >> ticketType of "**624**" (FailingCircuit), Colt will set 'issueSummary' to "Customer Operations / Circuit Up and Down" >> ticketType of "**1504**" (Information), Colt will set 'issueSummary' to "Service / Fault Report / Request for detailed Incident report >> ticketType of "**1014**" (Other)	
attachment	Complex Type	No	Base 64 Encoded Contents	
attachment/name	String	Yes	File Name	
attachment/content	Base 64 Encoded	Yes	File Content – Base 64 Encoded	
note	Complex Type	No	Ticket Comments	
note/text	String	Yes	Additional Comments	
relatedEntity	Complex Type	Yes	CircuitID / ServiceID Details	
relatedEntity/name	String	Yes	Subject that the ticket is being created on. Service/NEID in on Product tab of ticket (like Customer Circuit ID)	
relatedParty	Complex Type	Yes	CustomerAccountName	
relatedParty/name	String	Yes	CustomerAccountName	

Output Data (Asynch)	<table><tr><th>Name</th><th>Type</th><th>Mandatory</th><th>Description</th><th>Value</th></tr><tr><td>eventId</td><td>String</td><td>Yes</td><td>UUID</td><td></td></tr><tr><td>eventTime</td><td>String</td><td>No</td><td>Ticket Creation Response Time</td><td></td></tr><tr><td>eventType</td><td>String</td><td>Yes</td><td>Response Message Type</td><td>TroubleTicketCreateEvent</td></tr><tr><td>event</td><td>Complex</td><td>Yes</td><td></td><td></td></tr><tr><td>Event\troubleTicket</td><td>Complex</td><td>Yes</td><td></td><td></td></tr><tr><td>..\troubleTicket\id</td><td>Complex Type</td><td>No</td><td>Colt Ticket Number</td><td></td></tr><tr><td>..\troubleTicket\externalId</td><td>String</td><td>Yes</td><td>Customer Ticket Number</td><td></td></tr><tr><td>..\troubleTicket\Status</td><td>String</td><td>Yes</td><td>Status of the ticket</td><td>acknowledged</td></tr></table>
Input Sample	CreateTicket.txt
Output Synch Response / Acknowledgement	Please Refer the below Section "**Synch Response / Acknowledgement**"
Output Asynch Response Sample	CreateTicketResponse.txt
Communication Pattern	REST Asynch Request-Response

5.1.2 Update Ticket

Integration Scenario Name	**Update Ticket**
Interface	API Gateway
Recipients	Vendors
Transport	REST over HTTPS

Operation Name	PATCH
Description	This operation allows you to modify any field except the status and work information.
Input Data (parameters)	

Name	Type	Mandatory	Description	Value
id	String	Yes	Colt Ticket Number	
externalId	String	Yes	Customer TicketID	
ticketEscalation	String	No	Required only when there is an escalation, Field Value – 1 else 2 if there is no escalation.	
attachment	Complex Type	No	Base 64 Encoded Contents	
attachment/name	String	Yes	File Name	
attachment/content	Base 64 Encoded	Yes	File Content – Base 64 Encoded	
Attachments/FileData	Base 64 Encoded	Yes	File Content	
note	Complex Type	No	Ticket Comments	
note/text	String	Yes	Additional Comments	
relatedEntity	Complex Type	Yes	CircuitID / ServiceID Details	
relatedEntity/name	String	Yes	Subject that the ticket is being created on. Service/NEID in on Product tab of ticket (like Customer Circuit ID)	
relatedParty	Complex Type	Yes	CustomerAccountName	
relatedParty/name	String	Yes	CustomerAccountName	

Output Data (Asynch)	

Name	Type	Mandatory	Description	Value
eventId	String	Yes	UUID	
eventTime	String	No	Ticket Creation Response Time	
eventType	String	Yes	Response Message Type	TroubleTicketAttributeValueChangeEvent
description	String	No	Error Description	
event	Complex	Yes		
Event\troubleTicket	Complex	Yes		
.\troubleTicket\id	Complex Type	No	Colt Ticket Number	
.\troubleTicket\externalId	String	Yes	Customer Ticket Number	

Input Sample	UpdateTicket.txt
Output Synch Response / Acknowledgement	Please Refer the below Section "**Synch Response / Acknowledgement**"
Output Asynch Response Sample	UpdateTicketResponse.txt
Communication Pattern	REST asynch Request-Response

5.1.3 Verify Ticket Resolution

Integration Scenario Name	**Verify Ticket Resolution**
Interface	API Gateway
Recipients	Vendors
Transport	REST over HTTPS
Operation Name	PATCH

Description	The customer to send an acceptance or rejection of a ticket resolution to Colt shall use this operation.
Input Data (parameters)	***Verify Ticket Resolution Data Type***

Name	Type	Mandatory	Description	Value
id	String	Yes	Colt Ticket Number	
externalId	String	Yes	" Customer Ticket Number "	
status	String	Yes	Ticket Status	Customer maps as below: "resolved" for TicketResolutionAccpted "inProgress" for TicketResolutionRejected
statusChangeReason	String	Yes	Ticket Status Reason	Map "CloseOutVerification" value as: 1 - If "statusChangeReason" value is "TicketResolutionAccepted" 2 - If "statusChangeReason" value is "TicketResolutionRejected"
note	Complex Type	No	Ticket Comments	
note/text	String	Yes	Additional Comments	
relatedParty	Complex Type	Yes	CustomerAccountName	
relatedParty/name	String	Yes	CustomerAccountName	

Output Data

Name	Type	Mandatory	Description	Value
eventId	String	Yes	UUID	
eventTime	String	No	Ticket Creation Response Time	

eventType	String	Yes	Response Message Type	TroubleTicketAttributeValueChangeEvent
description	String	No	Error Description	
event	Complex	Yes		
Event\troubleTicket	Complex	Yes		
.\troubleTicket\id	Complex Type	No	Colt Ticket Number	
.\troubleTicket\externalId	String	Yes	Customer Ticket Number	

Input Sample	verifyTicketResolution.txt
Output Synch Response / Acknowledgement	Please Refer the below Section "**Synch Response / Acknowledgement**"
Output Asynch Response Sample	verifyTicketResolutionResponse.txt
Communication Pattern	REST async Request-Response

5.1.4 Cancel Ticket Resolution

Integration Scenario Name	**Cancel Ticket Resolution**
Interface	API Gateway
Recipients	Vendors
Transport	REST over HTTPS
Operation Name	POST
Description	The customer to send a reactive ticket cancellation request.
Input Data (parameters)	***Cancel Ticket Data Type***

Name	Type	Mandatory	Description	Value
id	String	Yes	Colt Ticket Number	

externalId	String	Yes	" Customer Ticket Number "	
status	String	Yes	Ticket Status	
statusChangeReason	String	Yes	Ticket Status Reason	
relatedParty	Complex Type	Yes	CustomerAccountName	
relatedParty/name	String	Yes	CustomerAccountName	

Output Data

Name	Type	Mandatory	Description	Value
eventId	String	Yes	UUID	
eventTime	String	Yes	Ticket Creation Response Time	
eventType	String	Yes	Response Message Type	
Description	String	Yes	Error Description	
event	Complex Type	Yes		
Event\troubleTicket	Complex Type	Yes		
..\troubleTicket\id	String	No	Colt Ticket Number	
..\troubleTicket\externalId	String	Yes	Customer Ticket Number	

Input Sample	CancelTicket.txt
Output Synch Response / Acknowledgement	Please Refer the below Section "**Synch Response / Acknowledgement**"
Output Asynch Response Sample	CancelTicketResponse.txt
Communication Pattern	REST async Request-Response

Synch Response / Acknowledgement

Please Note: For all the above reactive ticket to colt from the vendor there will be synch acknowledgement is sent as soon we receive the request, refer the below table.

Integration Scenario Name	**Synch Response**
Interface	API Gateway

Recipients	Vendor
Transport	REST over HTTPS
Operation Name	N/A
Description	Colt will send back an Acknowledgement/Response for the received Request from Customer.
Data (parameters)	***Synch Response***

Name	Type	Mandatory	Description	Value
Status	String	Yes	Response Type	
code	String	No	Error Code	
reason	String	No	Error Description	
message	String	No	Error Message	

Output Synch Response Sample	synchTicketAcknowledgement.txt

5.2 COLT Triggered Transactions

Integration Scenario Name	System Interaction Scenarios	Message Type(Elements)	Transactions Type
Proactive Ticket Notification	This transaction shall be used by Colt to send a proactive ticket request to the customer.	TroubleTicketCreateEvent	Outbound
Colt Comment Update	This transaction shall be used by Colt to send comments to the customer.	TroubleTicketAttributeValueChangeEvent	Outbound
Colt Sets Stop Clock for an Incident	This transaction shall be used by Colt to send the Clock Stop Update.	TroubleTicketStatusChangeEvent	Outbound
Colt Removes Stop Clock for an Incident	This transaction shall be used by Colt to send the removal Clock Stop Update.	TroubleTicketStatusChangeEvent	Outbound
Colt Cancels Incident	This transaction shall be used by Colt to send proactive ticket cancellation details to the customer.	TroubleTicketStatusChangeEvent	Outbound
Colt Resolves Incident	This transaction shall be used by Colt to send	TroubleTicketResolvedEvent	Outbound

	ticket resolution details to the customer.		
Colt Closes Incident	This transaction shall be used by Colt to send ticket closure details to the customer.	TroubleTicketStatusChangeEvent	Outbound

5.2.1 Proactive Ticket Notification

Integration Scenario Name	**Proactive Ticket Notification**
Interface	API GATEWAY
Recipients	Customers
Transport	REST over HTTPS
Operation Name	POST
Description	A new ticket is initiated by the Colt Incident management system (IMS) which creates a ticket request to be sent across to the customer. The ticket is received by the customer and a Confirmation Notification Receipt is initiated by the customer back to the Colt.
Output Data	***Proactive Ticket Notification***

Name	Type	Mandatory	Description	Value
eventId	String	Yes	Unique Identifier	
eventTime	String	Yes	Sysdatetime	
eventType	String	Yes	Map string "TroubleTicketCreateEvent" for Create Ticket	
event	Complex Type	Yes		
event\troubleTicket	Complex Type	Yes		
id	String	Yes	TroubleReportID – “Colt Ticket Number”	
creationDate	String	No	Ticket creation date	
description	String	Yes	Ticket description	
name	String	Yes	Name of the ticket or short description	
severity	String	No		
status	String	Yes	Set to “acknowledged”	
ticketType	String	Yes	Type of the ticket	
note	Complex Type	No	Additional comments	

note/text	String	Yes	Additional comments	
relatedEntity	Complex Type	Yes	CircuitID / ServiceDetails	
relatedEntity/name	String	Yes		

Communication Pattern	**REST Synch Request-Response**

5.2.2 Colt Comment Update

Integration Scenario Name	**Colt Adds Comment**
Interface	API GATEWAY
Recipients	Customers
Transport	REST over HTTPS
Operation Name	POST
Description	This transaction shall be used by Colt to send comments to the customer.
Output Data	Please refer "**5.3 Notify Output Data Sample**" Section
Output Sample	Please refer "**5.4 Notify Ticket Sample**" Section
Communication Pattern	REST sync Request-Response

5.2.3 Colt Sets Stop Clock for an Incident

Integration Scenario Name	**Colt Sets Stop Clock for an Incident**
Interface	API GATEWAY
Recipients	Customers
Transport	REST over HTTPS
Operation Name	POST
Description	This transaction shall be used by Colt to send the Clock Stop Update.
Output Data	Please refer "**5.3 Notify Output Data Sample**" Section
Output Sample	Please refer "**5.4 Notify Ticket Sample**" Section
Communication Pattern	REST sync Request-Response

5.2.4 Colt Removes Stop Clock for an Incident

Integration Scenario Name	**Colt Removes Stop Clock for an Incident**
Interface	API GATEWAY
Recipients	Customers
Transport	REST over HTTPS
Operation Name	POST
Description	This transaction shall be used by Colt to send the removal Clock Stop Update.
Output Data	Please refer "**5.3 Notify Output Data Sample**" Section
Output Sample	Please refer "**5.4 Notify Ticket Sample**" Section
Communication Pattern	REST sync Request-Response

5.2.5 Colt Cancels Incident

Integration Scenario Name	Colt Cancels Incident
Interface	API GATEWAY
Recipients	Customers
Transport	REST over HTTPS
Operation Name	POST
Description	This transaction shall be used by Colt to send proactive ticket cancellation details to the customer.
Output Data	Please refer "**5.3 Notify Output Data Sample**" Section
Output Sample	Please refer "**5.4 Notify Ticket Sample**" Section
Communication Pattern	REST sync Request-Response

5.2.6 Colt Resolves Incident

Integration Scenario Name	Colt Resolves Incident
Interface	API GATEWAY
Recipients	Customers
Transport	REST over HTTPS
Operation Name	POST
Description	This transaction shall be used by Colt to send ticket resolution details to the customer.
Output Data	Please refer "**5.3 Notify Output Data Sample**" Section
Output Sample	Please refer "**5.4 Notify Ticket Sample**" Section
Communication Pattern	REST sync Request-Response

5.2.7 Colt Closes Incident

Integration Scenario Name	Colt Closes Incident
Interface	API GATEWAY
Recipients	Customers
Transport	REST over HTTPS
Operation Name	POST
Description	This transaction shall be used by Colt to send ticket closure details to the customer.
Output Data	Please refer "**5.3 Notify Output Data Sample**" Section
Output Sample	Please refer "**5.4 Notify Ticket Sample**" Section
Communication Pattern	REST sync Request-Response

5.3 Notify Output Data Sample

Name	Type	Mandatory	Description	Value
eventId	String	Yes	Unique ID	

eventTime	String	Yes	sysdatetime	
eventType	String	No	Ticket Status	
event	String	No	TroubleTicketAttributeValueChangeEvent	
Event\troubleTicket	String	No	Colt Ticket Number	
.\troubleTicket\id	String	No	Customer Ticket Number	
.\troubleTicket\status	String	Yes	Ticket Status	
.\troubleTicket\statusChangeReason	String	No	Ticket Status Change Reason	
note	Complex Type	No	Ticket Comments	
note/text	String	Yes	Additional Comments	

5.4 Notify Ticket Sample

StatusNotify.txt

6 PRIORITY AND STATUS VALUES

6.1 Impact, Urgency & Priority Matrix

Customer Severity	Colt-ReactiveTicket	Colt-ProactiveTicket
Sev1	1-Critical	Sev1
Sev2	2-High	Sev2
Sev3	3-Medium	Sev3
Sev4	4-Low	Sev4

6.2 Status Values

Assigned
In Progress
Pending
Resolved
Closed
Cancelled

7 MAPPING SHEET

TMF621_TroubleTicket_ParameterMapping

8 TMF621 TROUBLE TICKET API SPECIFICATION

TMF621_Trouble_Ticket_API_REST_Speci

9 SWAGGER FILES

TMF621_TroubleTicket_Swagger.json