



REST SPECIFICATION TEMPLATE

Document Number: 1

Document Version: 1.0

Date: Aug, 2016

Document Status: Draft

NOTICE

Copyright © TeleManagement Forum 2013. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,
East Tower – 10th Floor,
Morristown, NJ 07960 USA
Tel No. +1 973 944 5100
Fax No. +1 973 944 5110

TM Forum Web Page: www.tmforum.org

TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

NOTICE.....	2
Table of Contents.....	3
Introduction.....	9
Program Specification Resources.....	10
Program Member Resources.....	10
SPECIFIC Considerations.....	11
eTOM.....	12
SAMPLE USE CASES.....	13
Use Case 1: Administration of Loyalty Program Specification Master Reference Data.....	13
Description.....	13
Main Actors.....	13
Use Case Steps.....	13
Example of API Usage in the Context of the Use Case.....	14
Success Outcome.....	14
Use Case 2: On-boarding of LoyaltyProgramMembers to specific LoyaltyProgramProduct.....	14
Description.....	14
Main Actors.....	14
Use Case Steps.....	14
Example of API Usage in the Context of the Use Case.....	15
Success Outcome.....	15
Use Case 3: Processing of events that may result in a LoyaltyAction being executed.....	15
Description.....	15
Main Actors.....	15
Use Case Steps.....	15
Example of API Usage in the Context of the Use Case.....	16
Success Outcome.....	16
Use Case 4: Burn points as part of a LoyaltyProgramMember CustomerPayment.....	16



Description 16

Main Actors 16

Use Case Steps 17

Example of API Usage in the Context of the Use Case 17

Success Outcome 17

Use Case 5: Specific Example - CustomerOrder resulting in redeemable loyalty points earnings 17

 Description 17

 Main Actors 17

 Use Case Steps 18

 Example of API Usage in the Context of the Use Case 18

 Success Outcome 18

RESOURCE MODEL 19

 MANAGED ENTITY AND TASK RESOURCE MODELS 19

 LoyaltyProgramProductSpec Resource 19

 LoyaltyRule Resource 20

 LoyaltyCondition Resource 22

 LoyaltyAction Resource 22

 LoyaltyEventType Resource 23

 LoyaltyProgramProduct Resource 23

 LoyaltyProgramMember Resource 24

 LoyaltyAccount Resource 26

 LoyaltyBalance Resource 27

 LoyaltyEarn Resource 28

 LoyaltyBurn Resource 29

 LoyaltyEvent Resource 29

 LAYER RESOURCE 30

 LoyaltyProgramProductSpec Resource 30

 LoyaltyRule Resource 31

 LoyaltyCondition Resource 32



LoyaltyAction Resource	33
LoyaltyEventType Resource	34
LoyaltyProgramProduct Resource	34
LoyaltyProgramMember Resource	35
LoyaltyAccount Resource	36
LoyaltyBalance Resource	37
LoyaltyEarn Resource	38
LoyaltyBurn Resource	38
LoyaltyEvent Resource	39
Event Models	40
loyaltyProgramMemberCreateNotification	40
loyaltyProgramMemberUpdateNotification	40
loyaltyProgramMemberDeleteNotification	42
loyaltyProgramMemberProductCreateNotification	44
loyaltyProgramMemberProductUpdateNotification	45
loyaltyProgramMemberProductDeleteNotification	46
loyaltyEarnNotification	47
loyaltyBurnNotification	48
loyaltyEventNotification	48
API OPERATION TEMPLATES	50
GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}	50
PUT /loyaltyManagement/loyaltyProgramProductSpec/{ID}	52
PATCH /loyaltyManagement/loyaltyProgramProductSpec/{ID}	52
POST /loyaltyManagement/loyaltyProgramProductSpec	54
DELETE /LOYALTYMANAGEMENT/LOYALTYPROGRAMPRODUCTSPEC/{ID}	55
GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}	56
PUT /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}	57
PATCH /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}	58
POST /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule	60



DELETE /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID} 62

GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID} 63

PUT /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID} 64

PATCH /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID} 64

POST /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID} 64

DELETE /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID} 65

GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventType/{ID} 66

PUT /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventType/{ID} 67

PATCH /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventType/{ID} 67

POST /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventType 67

DELETE /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventType/{ID} 68

GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID} 69

PUT /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID} 70

PATCH /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID} 70

POST /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction 70

DELETE /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID} 71

GET /loyaltyManagement/loyaltyCondition/{ID} 72

PUT /loyaltyManagement/loyaltyCondition/{ID} 73

PATCH /loyaltyManagement/loyaltyCondition/{ID} 74

POST /loyaltyManagement/loyaltyCondition 75

DELETE /loyaltyManagement/loyaltyCondition/{ID} 76

GET /loyaltyManagement/loyaltyAction/{ID} 77

PUT /loyaltyManagement/loyaltyAction/{ID} 78

PATCH /loyaltyManagement/loyaltyAction/{ID} 80

POST /loyaltyManagement/loyaltyAction 81

DELETE /loyaltyManagement/loyaltyAction/{ID} 83

GET /loyaltyManagement/loyaltyEventType/{ID} 84

PUT /loyaltyManagement/loyaltyEventType/{ID} 85

PATCH /loyaltyManagement/loyaltyEventType/{ID} 86



POST /loyaltyManagement/loyaltyEventType87

DELETE /loyaltyManagement/loyaltyEventType/{ID}88

GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}88

PUT /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}90

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}90

POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct91

DELETE /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}94

GET /loyaltyManagement/loyaltyProgramMember/{ID}94

PUT /loyaltyManagement/loyaltyProgramMember/{ID}97

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}97

POST /loyaltyManagement/loyaltyProgramMember100

DELETE /loyaltyManagement/loyaltyProgramMember/{ID}101

GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount102

PUT /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}103

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}103

POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount103

DELETE /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}105

GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance105

PUT /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}108

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}108

POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance110

DELETE /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}112

GET /loyaltyManagement/loyaltyProgramMember/{ID} /loyaltyBalance/{ID}/loyaltyEarn113

POST /loyaltyManagement/loyaltyProgramMember/{ID} /loyaltyBalance/{ID}/loyaltyEarn114

GET /loyaltyManagement/loyaltyProgramMember/{ID} /loyaltyBalance/{ID}/loyaltyBurn115

POST /loyaltyManagement/loyaltyProgramMember/{ID} /loyaltyBalance/{ID}/loyaltyBurn117

POST /loyaltyManagement/loyaltyEvent118

API NOTIFICATION TEMPLATES120

Register LISTENER POST /loyaltyManagement/loyaltyProgramMember/hub120



Unregister LISTENER DELETE /loyaltyManagement/loyaltyProgramMember/hub/{id} 120

Publish loyaltyProgramMember POST /listener 121

Register LISTENER POST /loyaltyManagement/loyaltyProgramMemberProduct/hub 122

Unregister LISTENER DELETE /loyaltyManagement/loyaltyProgramMemberProduct/hub/{id} 122

Publish loyaltyProgramMemberProduct POST /listener 123

Register LISTENER POST /loyaltyManagement/loyaltyEarn/hub 124

Unregister LISTENER DELETE /loyaltyManagement/loyaltyEarn/hub/{id} 124

Publish loyaltyEarn POST /listener 125

Register LISTENER POST /loyaltyManagement/loyaltyBurn/hub 125

Unregister LISTENER DELETE /loyaltyManagement/loyaltyBurn/hub/{id} 126

Publish loyaltyBurn POST /listener 126

Register LISTENER POST /loyaltyManagement/loyaltyEvent/hub 127

Unregister LISTENER DELETE /loyaltyManagement/loyaltyEvent/hub/{id} 127

Publish loyaltyEvent POST /listener 128

OPEN ISSUES 130

Release History 131

INTRODUCTION

The TMForum Loyalty Management Open API is documented in this specification. The Loyalty Management API specification references the SID Release 14.5.1 and the Loyalty API R14.0 specification.

The Loyalty API supports the management of loyalty program specifications, loyalty program members, their associated products and loyalty accounts with loyalty balances. The scope of the API also covers the management of loyalty rules and under what conditions the associated loyalty actions must be applied.

The TMForum Loyalty Program API addresses an area of Communication Service Provider functionality in which a high degree of differentiation is required to set CSPs apart from their competitors. The question can therefore be posed about the merits of standardisation in this area. The API aims to provide a standardised integration interface for the definition of Loyalty Programs and the receipt of events from integrated systems in order to drive integration standardisation only. Loyalty Program differentiation will be achieved in program implementation. For example, the BusinessInteraction loyalty execution points represent internal and also program coalition partner business interactions which is seen as one of the main Loyalty Program differentiators.

As depicted in Figure 1, the Loyalty Management API addresses resources that can be classified into two broad categories outlined below

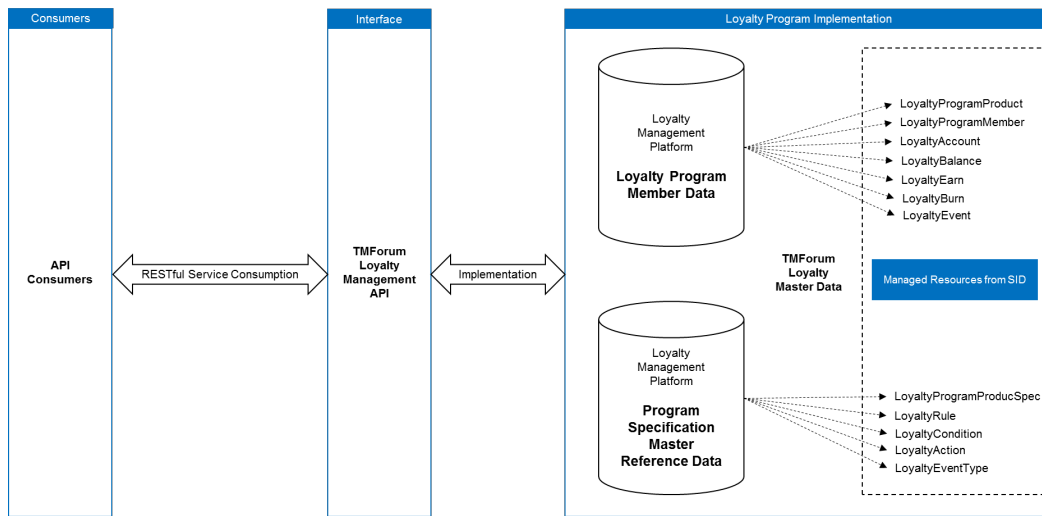


Figure 1 – Loyalty Program Managed Resource Context

Program Specification Resources

This grouping of resources define the Loyalty Program characteristics and therefore defines the program reference data.

This grouping represents resources that collectively determine loyalty program business rules and event processing behaviour, where events could be triggered by a variety of sources (consumers) that may result in a loyalty action (a points earn, business interaction or customer order action).

Program Member Resources

This grouping of managed resources is related to a Loyalty Program member, e.g. Loyalty Program members, loyalty program products to which Loyalty Program Members are signed up, accounts, balances, earn, burn and other events.

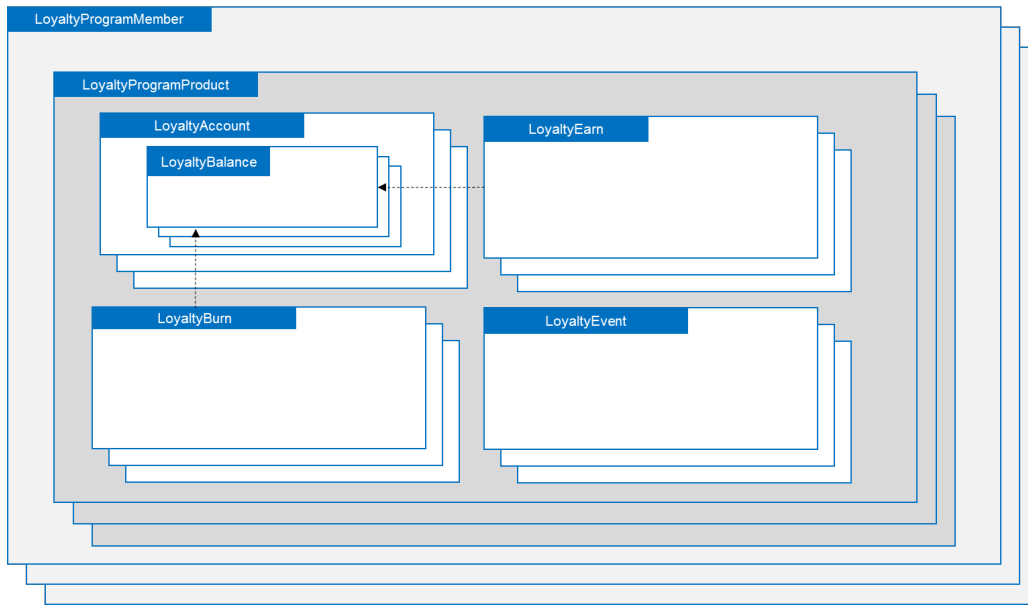


Figure 1 –LoyaltyProgramMember resource and contained resources

As depicted in Figure 1, the Loyalty Management API aims to support a per loyalty program member resource model as follows:

- A LoyaltyProgramMember opts in to one or more LoyaltyProgramProducts;
- A LoyaltyProgramProduct can contain one or more LoyaltyAccounts. While this is shown as containment in this diagram, multiple LoyaltyProgramProducts may be associated with the same LoyaltyAccount;
- A LoyaltyAccount can contain multiple LoyaltyBalances;
- LoyaltyEarm and LoyaltyBurn can be viewed as special event types that affect LoyaltyBalance with credit and debit transaction entries respectively. These will typically be reflected on the Loyalty Program BSS platform as specific credit and debit event data records;

- LoyaltyEvent is an unmanaged resource¹ but represents a variety of loyalty program event triggers received from external sources, identified by an event type specification. A LoyaltyEvent can specify event types such as Order Creation, usage events and others. LoyaltyEvent is, however, shown in in Figure 1 as it will be presented in the Loyalty Program Platform as an event data record under the LoyaltyProgramMember profile to serve as audit trail of the event. The event may result in subsequent LoyaltyActions that may, in turn, result in further event data records to be logged to record the corresponding benefits allocated as a result of the event.

SPECIFIC Considerations

As Loyalty Programs are established and grow in terms of sophistication, the number and variety of events processed will inevitably increase as will the number of LoyaltyRules. For example, social media and customer sentiment integration derived from unstructured data sources may be added. This means that the API should support fine-grained association of LoyaltyEvents with LoyaltyRules, i.e. each event received should not be evaluated against each LoyaltyRule. For this reason, this specification proposes the introduction of a LoyaltyEventType SID entity that enables the association of a kind of LoyaltyEvent which we perceive as an unmanaged resource, by using the eventType name value pair, with a subset of LoyaltyRules association with a LoyaltyProgramProductSpec.

Another key consideration is that of association management between LoyaltyProgramMembers. For example, if the CSP customer is a multi-entity corporate customer, many subsidiaries in the company's ERP company hierarchy may want to benefit from the same Loyalty Program. This means that multiple LoyaltyProgramMembers may want to earn and burn from the same LoyaltyAccount and its underlying LoyaltyBalances. The authors were uncertain at the time of writing of this draft if an association relationship is provided for within the present SID model. Regardless, association management between LoyaltyProgramMembers will be a requirement for Loyalty Program management.

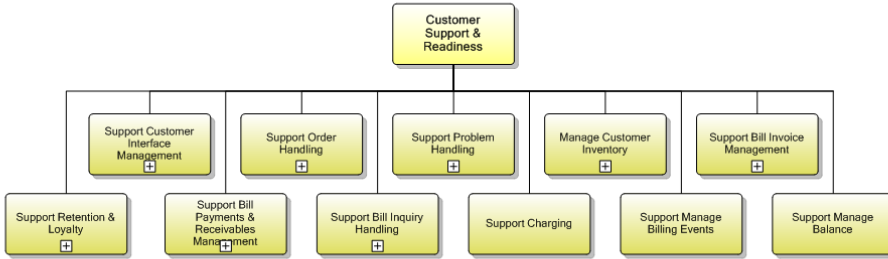
The audience is also referred to the Open Issues section at the end of this document for other open issues to be addressed.

¹ Proposed change in SID to augment LoyaltyRule with LoyaltyEventType to enable finer-grained mapping of events to application LoyaltyRules

eTOM

Loyalty and Customer Retention forms part of the Customer Support & Readiness part of eTOM.

1.3.1 Customer Support & Readiness



This was created from the Framework 16.0 Model

1.3.1.4 Support Retention & Loyalty

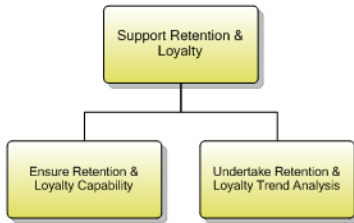


Diagram Description

Ensure that all information, materials, systems and resources are available so that the Retention & Loyalty processes can be completed without delay, when a request is received from a customer

This was created from the Framework 16.0 Model

SAMPLE USE CASES

The following mainstream use cases are provided to enhance understanding of the underlying API use in support of the Use Cases:

- i. Administration of Loyalty Program Specification Master Reference Data
 - a. Creation of LoyaltyProgramProductSpecification
 - b. Creation of one or more LoyaltyRules
 - c. Creation of one or more LoyaltyConditions
 - d. Creation of Loyalty Actions
 - e. Creation of Loyalty Event Types
- ii. On-boarding of LoyaltyProgramMembers to specific LoyaltyProgramProduct
- iii. Processing of events that may result in a LoyaltyAction being executed (LoyaltyEarn, CustomerOrder, BusinessInteraction)
- iv. Burn points as part of a LoyaltyProgramMember CustomerPayment

Use Case 1: Administration of Loyalty Program Specification Master Reference Data

Description

The main purpose of this use case is the administration of all reference data required to process events for the purpose of loyalty benefit processing in line with this defined reference data.

Main Actors

- A Loyalty Program Administrator
- A User Interface using the Loyalty Management API
- A BSS platform capable of storing the Loyalty Program reference data

Use Case Steps

- i. The Loyalty Program Administrator creates a new loyalty program specification in line with business requirements, e.g. a program to capture the youth market;
- ii. The Loyalty Program Administrator creates one or more business rules applicable to the program, e.g. that only program members between the age of 18 and 21 qualify for the program;
- iii. The Loyalty Program Administrator specifies the set of conditions applicable to each Loyalty Rule, e.g. minimum age = 18 and maximum age = 21;
- iv. The Loyalty Program Administrator specifies the type of loyalty action and the execution point responsible for fulfilling the loyalty benefits, e.g. earn 100 points on placement of customer order.
- v. The Loyalty Program Administrator defines the qualifying event types that should trigger the loyalty rule with its underlying conditions and actions, e.g. only customer order events should be processed.

Example of API Usage in the Context of the Use Case

Corresponding with the numbering in the use case steps, the following API interactions support the use case:

- i. The User Interface consumes the LoyaltyProgramProductSpec resource to create the new program specification;
- ii. The User Interface consumes the LoyaltyRule resource to create one or more rules related to the newly created LoyaltyProgramProductSpec;
- iii. The User Interface consumes the LoyaltyCondition resource to create one or more conditions to specify whether or not the LoyaltyRule is applicable to a specific event. After creation, the condition must be linked to the LoyaltyRule;
- iv. The User Interface consumes the LoyaltyAction resource to create one or more actions that describe the fulfillment of the loyalty execution. After creation, the action must be linked to the LoyaltyRule;
- v. The User Interface consumes the EventType resource to define event types that trigger specific LoyaltyRules to be evaluated upon receipt of events of the type specified. After creation, the event type must be linked to the LoyaltyRule;

Success Outcome

After completion of these API interactions, a complete Loyalty Program business rule has been created that is referenced as part of the event processing pattern to definitively allocate loyalty benefits based on this metadata.

Use Case 2: On-boarding of LoyaltyProgramMembers to specific LoyaltyProgramProduct

Description

The main purpose of this use case is to successfully onboard new Loyalty Program Members for the loyalty program (LoyaltyProgramProductSpec).

Main Actors

- Customer: An existing customer (PartyRole)
- Access Channel System: An external system that assists a customer of the CSP with an authenticated customer interaction with the CSP and where the opportunity to enrol for the Loyalty Program is presented. This could be an online portal, a CRM system or other.
- Loyalty Platform: A BSS platform supporting the Loyalty Program functions

Use Case Steps

- i. The customer interacts with the CSP using an Access Channel System and the customer is authenticated as part of this interaction;
- ii. As part of an authenticated journey, the customer is presented with the option to sign up for Loyalty Program and enrolls for the Loyalty Program.

Example of API Usage in the Context of the Use Case

Corresponding with the numbering in the use case steps, the following API interactions support the use case:

- i. As part of customer authentication, the access channel system can use the LoyaltyProgramProduct API resource to verify if the customer is already opted in to the Loyalty Program.
- ii. In response to the customer electing to sign up for the loyalty program, the API LoyaltyProgramMember CREATE operation is used to record the program enrolment. As part of the implementation of the API call, the following activities will take place:
 - a. The Loyalty Platform will record the specific program that the enrolment applies to (i.e. the corresponding LoyaltyProgramProductSpec) and this record represents the LoyaltyProgramProduct as an instance of Loyalty Program management for the customer;
 - b. The Loyalty Platform can also create a Loyalty Account as part of the program enrolment and perhaps corresponding opening balances.

Success Outcome

After completion of these API an existing customer (PartyRole) is successfully enrolled for a Loyalty Program (as identified by the corresponding LoyaltyProgramProductSpec). The customer is now a LoyaltyProgramMember and the specific LoyaltyProgramProductSpec subscribed to has been recorded by the LoyaltyProgramProduct instance.

Use Case 3: Processing of events that may result in a LoyaltyAction being executed

Description

The main purpose of this use case is to process externally originating event triggers received from external sources that represents an event of relevance to the management of a Loyalty Program.

Main Actors

- **External OSS/BSS Platform:** Any external OSS/BSS platform that emits events that may be of relevance for Loyalty Program processing and that will be received by the event handler
- **Loyalty Platform:** A BSS platform supporting the Loyalty Program functions.

Use Case Steps

- i. An event is received by means of a POST operation to the loyaltyEvent resource;
- ii. The event handling implementation sources the event type identifier (using the eventType Name Value Pair) and determines the qualifying subset of all Loyalty Rules to be considered for loyalty benefit processing;
- iii. The Loyalty API implementation classes or the Loyalty Platform implements the following:
 - a. For each qualifying Loyalty Rule, the associated conditions are evaluated against the Loyalty Event information received. At this point of evaluation the event against the identified Loyalty Rule, the Loyalty Program Member profile information would typically already have been sourced to allow the aggregated event, loyalty member profile and Loyalty Program master reference data to be used in the processing of the conditions associated with Loyalty Rules.
 - b. For each matching condition, the configured loyalty actions are performed to implement the desired loyalty programme effect.

Example of API Usage in the Context of the Use Case

Corresponding with the numbering in the use case steps, the following API interactions support the use case:

- i. The event handling implementation receives the event triggered by an external BSS/OSS platform ;
- ii. The event handling implementation or Loyalty Platform obtains all EventType resources defined as well as their links to LoyaltyRules;
- iii. The event handling implementation or Loyalty Platform applies the conditions associated with each LoyaltyRule and determines if the LoyaltyRule should be applied. In the event that the LoyaltyRule is determined to be applicable based on the conditions, the associated LoyaltyActions for the LoyaltyRule is obtained and the associated LoyaltyExecutionPoints that the LoyaltyAction implements are consumed to implement the desired real-world effect. The real-world effect may be one of a LoyaltyEarn, CustomerOrder or a BusinessInteraction. The Loyalty Platform will be responsible for the fulfilment of the desired real-world effect.

Success Outcome

After completion of the event processing cycle, the LoyaltyProgramMember impacted by the event will have received a benefit in the form of one or more of:

- LoyaltyEarn – a loyalty balance was credited in accordance with the benefits determined by means of fulfilment of a LoyaltyAction;
- CustomerOrder – a customer order was placed with a business entity (internal or a coalition partner) for a product;
- BusinessInteraction – a qualifying LoyaltyAction resulted in an interaction with a business unit (internal) or with an external business, e.g. a Loyalty Program coalition partner, to implement a loyalty benefit fulfilled by an organisational division or external business partner.

As many LoyaltyActions may be associated with each qualifying LoyaltyRule and as a single event may trigger multiple rules, the real-world effect may be that combinations of benefits, e.g. **a LoyaltyEarn as well as a BusinessInteraction, may result from the processing of a single event received.**

Use Case 4: Burn points as part of a LoyaltyProgramMember CustomerPayment

Description

The main purpose of this use case is to describe the process whereby a LoyaltyProgramMember can use earned Loyalty Points balance available for product or service payment either using it as payment resource in full or for partial payment.

Main Actors

- **LoyaltyProgramMember:** An existing customer (PartyRole) that have previously opted in to a Loyalty Program and that has accrued redeemable loyalty currency.
- **External OSS/BSS Platform:** Any external OSS/BSS platform that emits events that may be of relevance for Loyalty Program processing and that will be received by the event handler
- **Loyalty Platform:** A BSS platform supporting the Loyalty Program functions.

Use Case Steps

- i. A Loyalty Program Member interacts with the CSP customer through an authorised channel (e.g. Online Portal) and is authenticated;
- ii. The Loyalty Program Member elects to purchase a product or service offer from the CSP;
- iii. The customer is presented in the shopping basket that redeemable Loyalty Balance is available that may be used towards the payment for the product or service;
- iv. The customer elects to use redeemable loyalty balance towards the payment to pay either in part or in full for the product or service. The customer proceeds to payment and pays with loyalty earnings either as part or as full payment.

Example of API Usage in the Context of the Use Case

Corresponding with the numbering in the use case steps, the following API interactions support the use case:

- i. The authorised channel system (external OSS/BSS platform) obtains the profile of the LoyaltyProgramMember by performing a GET operation on the LoyaltyProgramMember resource. This operation returns whether or not the customer is part of a Loyalty Program and also returns present account and balance information;
- ii. No interaction with the Loyalty Management API takes place in this step;
- iii. No interaction with the Loyalty Management API takes place in this step;
- iv. A POST operation is done to the LoyaltyBurn resource to record the loyalty earnings spend event. The Loyalty Platform emits a loyaltyBurnEvent to notify external platforms subscribed to Loyalty Management events about the burn event.

Success Outcome

After completion of the customer purchase transaction, the corresponding loyalty balance will have been debited with the corresponding loyalty currency used by the LoyaltyProgramMember towards the purchase. External subscribers would have been notified about the burn event through the event notification. It would also be anticipated that the Loyalty Platform would log an event data record (audit trail) representing a customer payment and corresponding records for the debit transaction performed on the loyalty account balance.

Use Case 5: Specific Example - CustomerOrder resulting in redeemable loyalty points earnings

Description

The main purpose of the Loyalty API is to retain customers. Customers are typically awarded loyalty earnings for the following reasons:

- Seniority in terms of customer relationship
- The importance of revenue for the customer
- Reducing the risk of churn.

Main Actors

- **External OSS/BSS Platform:** Any external OSS/BSS platform that emits events that may be of relevance for Loyalty Program processing and that will be received by the event handler
- **Loyalty Platform:** A BSS platform supporting the Loyalty Program functions.

Use Case Steps

A typical sequence is as follows:

- A customer places an order with the CSP and an external BSS platform is used to process the order, e.g. a CRM system;
- An event is emitted by the external BSS platform that the Loyalty Platform is subscribed to and the Loyalty Platform receives this event;
- The event is processed following the process defined in use case 3. As a result of this processing, the customer receives a Loyalty Earnings benefit.

Example of API Usage in the Context of the Use Case

- The Loyalty API registers a listener on any event that may trigger a loyalty event, e.g. CustomerOrder, BillingAccount and Product usage.
- When the event (that is listened on by the Loyalty API) occurs, the LoyaltyEvent resource is called.
- The LoyaltyEvent finds all the LoyaltyProgramProduct elements with the same type as the event. For this use case, all the LoyaltyProgramProduct of type 'CustomerOrder' is found.
- The LoyaltyProgramProduct associated LoyaltyRules are found. If the LoyaltyProgramMember (as provided by the event that triggered the loyalty event notification) has an instance of a LoyaltyProgramProduct, the LoyaltyRule LoyaltyConditions are evaluated.
- If all the LoyaltyConditions or some LoyaltyConditions (depending on the isCNF attribute) evaluate to true, the associated LoyaltyExecutionPoints are executed. The associated LoyaltyAction describes the LoyaltyExecutionPoint attributes.
- The LoyaltyAction, in this scenario, is a LoyaltyEarn transaction.
- A Loyalty Earn notification is emitted and subscribers to this event are notified about the earning event.

Success Outcome

After processing of the customer order, the real-world effect is that the appropriate loyalty account balance will have been credited with the loyalty earnings amount. It is expected that the Loyalty Platform will have logged event data records to record the LoyaltyBalance credit and that this record will form part of a LoyaltyProgramMember's earnings audit trail history.

RESOURCE MODEL

MANAGED ENTITY AND TASK RESOURCE MODELS

LoyaltyProgramProductSpec Resource

The loyalty program product specification resource is the root entity for product specification management.

A loyalty program product specification is a detailed description of a loyalty program made available externally in the form of a LoyaltyProduct to LoyaltyProgramMembers.

A LoyaltyProgramProductSpec defines one or more LoyaltyRules that have to be checked in order to identify the actions to apply.

Resource IDs for loyalty program product specifications are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyProgramProductSpec resource in JSON format.

```
{
  "id": "121",
  "name": "UpComingProfessionalsProgram",
  "description": "Loyalty Program to ensure that prepaid youth market is retained",
  "productNumber": "983284",
  "brand": "Globetom",
  "needsLoyaltyAccount": true,
  "lifeCycleStatus": "suspended",
  "href": "/loyaltyManagement/loyaltyProgramProductSpec/121",
  "validFor": {
    "endDateTime": "2016-12-31T23:59:59Z",
    "startDateTime": "2016-01-01T00:00:00Z"
  },
  "loyaltyRule": [{
    "id": "1",
    "name": "YouthRule",
    "description": "Verify if the customers age qualifies for youth program benefits",
    "isCNF": true,
    "hasSubRules": false,
    "isMandatoryEvaluation": true,
    "usage": "Subscribers younger than 23."
  }
  ]
}
```

```

"keywords": "age,youth",
"policyName": "Age less than 23",
"href": "/loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1",
"loyaltyEventType": [{
  "id": "3",
  "eventType": "orderCreationNotification"
}],
"loyaltyCondition": [{
  "id": "1",
  "attribute": "age",
  "operator": "<",
  "value": "23",
  "href": "/loyaltyManagement/loyaltyCondition/1"
}],
"loyaltyAction": [{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "href": "/loyaltyManagement/loyaltyAction/111",
  "loyaltyExecutionPoint": {
    "commonName": "Earn50",
    "description": "",
    "action": "POST",
    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
      loyaltyAccount/{accountId}/loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "2.0"
  },
}
]}
}

```

LoyaltyRule Resource

A LoyaltyRule specifies:

- Events triggering the evaluation of the rule (LoyaltyEventType).
- Conditions that have to be checked (LoyaltyCondition).
- Actions that should be performed if the condition clause is evaluated to TRUE (LoyaltyAction).

Loyalty rules may result in either rewards directly used / usable by the customer or loyalty earns that are gathered on an account to be used to later pay a ProductOffering.

Resource IDs for loyalty rules are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyRule resource in JSON format.

```
{
  "id": "1",
  "commonName": "YouthRule",
  "description": "Verify if the customers age qualifies for youth program benefits",
  "isCNF": true,
  "hasSubRules": false,
  "isMandatoryEvaluation": true,
  "usage": "Subscribers younger than 23.",
  "keywords": "age,youth",
  "policyName": "Age less than 23",
  "href": "/loyaltyManagement/loyaltyProgramProductSpec/121/LoyaltyRule/1",
  "loyaltyEventType": [{
    "id": "3",
    "eventType": "orderCreationNotification"
  }],
  "loyaltyCondition": [{
    "id": "1",
    "attribute": "age",
    "operator": "<",
    "value": "23",
    "href": "/loyaltyManagement/loyaltyCondition/1"
  }],
  "loyaltyAction": [{
    "id": "111",
    "type": "LoyaltyEarn",
    "actionAttributes": {
      "quantity": 50
    },
    "href": "/loyaltyManagement/loyaltyAction/111"
  }],
  "loyaltyExecutionPoint": {
    "commonName": "Earn50",
    "description": "",
    "action": "POST",
  }
}
```

```
"endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
loyaltyAccount/{accountId}/loyaltyBalance/{balanceId}/loyaltyEarn",
"version": "2.0"
},
]]
}
```

LoyaltyCondition Resource

A LoyaltyCondition specifies the condition that has to be evaluated to decide if the LoyaltyAction(s) from the LoyaltyRule have to be performed.

Resource IDs for loyalty conditions are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyCondition resource in JSON format.

```
{
  "id": "1",
  "attribute": "age",
  "operator": "<",
  "value": "30",
  "href": "/loyaltyManagement/loyaltyCondition/1"
}
```

LoyaltyAction Resource

A LoyaltyAction is an action that needs to be performed if the loyalty rule's condition clause evaluates to TRUE. A LoyaltyAction may correspond either to a CustomerOrder (for example 100 SMS free), or to a BusinessInteraction (for example an SMS notifying the 100 SMS free) or to a LoyaltyEarn (for example 100 points on the LoyaltyAccount).

A LoyaltyAction describes a LoyaltyExecutionPoint. A LoyaltyExecutionPoint is a type of PolicyExecutionPoint. A LoyaltyExecutionPoint tracks the application of a LoyaltyAction and may trigger BusinessInteraction, CustomerOrder or LoyaltyEarn, depending on the type of the LoyaltyAction it belongs to.

Resource IDs for loyalty actions are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyAction resource in JSON format.

```
{
```

```

{id": "111",
"type": "LoyaltyEarn",
"actionAttributes": {
  "quantity": 50
},
"loyaltyExecutionPoint": {
  "commonName": "Earn50",
  "description": "Earn loyalty points",
  "action": "POST",
  "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/loyaltyAccount/
    {accountId}/loyaltyBalance/{balanceId}/loyaltyEarn",
  "version": "2.0"
},
"href": "/loyaltyManagement/loyaltyAction/111"
}

```

LoyaltyEventType Resource

A LoyaltyEventType is a qualifier specifying which incoming event types are associated with which LoyaltyRules.

The different types of LoyaltyEvent may be events that occurs on BillingAccount (Ex: amount of Invoice, recurring charge), on PartyRole (Ex: CustomerOrder amount), on Products (Ex: monthly international usage volume). The LoyaltyEventType resource limits the evaluation of loyalty rules by event type.

Resource IDs for loyalty event types are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyEventType resource in JSON format.

```

{
  "id": "111",
  "eventType": "orderCreationNotification"
}

```

LoyaltyProgramProduct Resource

A LoyaltyProgramProduct is described by a LoyaltyProgramProductSpec and represents an instance of the LoyaltyProgramProductSpec for a specific LoyaltyProgramMember.

If the needsLoyaltyAccount attribute in the LoyaltyProgramProductSpec that describes the LoyaltyProgramProduct is set to TRUE, then the LoyaltyProgramProduct will have a link to a LoyaltyAccount that it logs results to.

Resource IDs for loyalty program products are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyProgramProduct resource in JSON format.

```
{
  "id": "1211",
  "name": "DataUsageBenefit",
  "description": "Data Usage Loyalty Benefits",
  "productStatus": "suspended",
  "productSerialNumber": "S2345666",
  "href": "/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211",
  "validFor": {
    "endDateTime": "2017-12-19T16:42:20Z",
    "startDateTime": "2017-05-19T16:42:20Z"
  },
  "loyaltyProgramProductSpec": {
    "id": "121",
    "href": "/loyaltyManagement/loyaltyProgramProductSpec/121"
  },
  "loyaltyAccount": {
    "id": "1",
    "href": "/loyaltyManagement/loyaltyProgramMember/2222/loyaltyAccount/1"
  }
}
```

LoyaltyProgramMember Resource

A LoyaltyProgramMember is a type of PartyRole with rights to a LoyaltyAccount by means of a LoyaltyProgramProduct. The loyalty program member may earn or burn loyalty through the LoyaltyEarn and LoyaltyBurn actions.

Resource IDs for loyalty program members are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyProgramMember resource in JSON format.

```
{
  "id": "104",
  "href": "/loyaltyManagement/loyaltyProgramMember/JDSU778DS",
  "status": "active",
  "name": "Jane Joe",
  "validFor": {
    "startDateTime": "2015-04-19T16:42:23.0Z",
    "endDateTime": "2016-04-19T16:42:23.0Z"
  }
}
```



```

    },
    "loyaltyAccount": [{
      "id": "LoyaltyFund",
      "href": "/loyaltyManagement/loyaltyProgramMber/104/loyaltyAccount/LoyaltyFund",
      "loyaltyBalance": {
        "id": "iTunes",
        "balance": 1020,
        "validFor": {
          "startDateTime": "2015-04-19T16:42:23.0Z",
          "endDateTime": "2016-04-19T16:42:23.0Z"
        },
        "unit": "magicPoints",
        "loyaltyEarn": {
          "quantity": 1000,
          "dateTime": "2016-04-19T16:42:23.0Z",
          "openingBalance": 800,
          "closingBalance": 200
        },
        "loyaltyBurn": []
      }
    ]
  }],
  "loyaltyProgramProduct": {
    "id": "1211",
    "name": "DataUsageBenefit",
    "description": "Data Usage Loyalty Benefits",
    "productStatus": "suspended",
    "productSerialNumber": "S2345666",
    "loyaltyProgramProductSpec": {
      "id": "121",
      "href": "/loyaltyManagement/loyaltyProgramProductSpec/121"
    },
    "loyaltyAccount": {
      "id": "LoyaltyFund",
      "href": "/loyaltyManagement/loyaltyProgramMember/JDSU778DS/loyaltyAccount/LoyaltyFund"
    },
    "validFor": {
      "endDateTime": "2015-05-19T16:42:20Z",
      "startDateTime": "2017-05-19T16:42:20Z"
    }
  }
}

```

```

    },
    "href": "/loyaltyManagement/loyaltyProgramMember/JDSU778DS/loyaltyProgramProduct/1211"
  ]]
}

```

LoyaltyAccount Resource

A LoyaltyAccount corresponds to a set of balances to carry loyalty movement according to different valid period and unit.

A LoyaltyAccount may be required to realize a LoyaltyProgramProduct according to the rules carried by the corresponding LoyaltyProgramProductSpec. This is defined by the needsLoyaltyAccount attribute on the LoyaltyProgramProductSpec entity.

A LoyaltyAccount may collect results from one or more LoyaltyProgramProduct resources.

IDs for loyalty accounts are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyAccount resource in JSON format.

```

{
  "id": "ValueBundle",
  "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/ValueBundle",
  "loyaltyProgramProduct": {
    "id": "1211",
    "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyProgramProduct/1211"
  },
  "loyaltyBalance": [{
    "id": "iTunes",
    "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/ValueBundle/loyaltyBalance/iTunes",
    "unit": "NZD",
    "balance": 300,
    "validFor": {
      "startDateTime": "2016/02/19 18:42:23",
      "endDateTime": "2018/12/30 17:42:23"
    }
  }
]}

```

```
}

```

LoyaltyBalance Resource

A LoyaltyBalance is credited by LoyaltyEarn transactions and debited by LoyaltyBurn transactions.

A LoyaltyBalance belongs to a LoyaltyAccount.

IDs for loyalty balances are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyBalance resource in JSON format.

```
{
  "id": "iTunes",
  "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes",
  "unit": "NZD",
  "balance": 300.00,
  "loyaltyAccount": {
    "id": "ValueBundle",
    "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/ValueBundle "
  }
  "validFor": {
    "startDateTime": "2016/02/19 18:42:23",
    "endDateTime": "2018/12/30 17:42:23"
  },
  "loyaltyEarn": [{
    "id": "843G-838F-HY23-0238",
    "quantity": 20.00,
    "openingBalance": 280.00,
    "closingBalance": 300.00,
    "dateTime": "2016-07-29 12:18:51",
    "description": "A loyalty event triggered and points earned."
  }],
  {
    "id": "146G-3408-WW40-P238",
    "quantity": 10.00,
    "openingBalance": 250.00,
    "closingBalance": 260.00,
  }
}
```

```

    "dateTime": "2016-07-12 23:34:25",
    "description": "Earn loyalty points on Justin Bieber album purchase."
  }],
  "loyaltyBurn": [{
    "id": "94JU-03J8-57S4-0893",
    "quantity": 20.00,
    "openingBalance": 1000.00,
    "closingBalance": 1020.00,
    "dateTime": "2016-07-08 13:11:27",
    "description": "Beyonce CD purchase loyalty points."
  }],
  {
    "id": "0347-4378-HT82-6CSJ",
    "quantity": 179.00,
    "openingBalance": 3430.00,
    "closingBalance": 3251.00,
    "dateTime": "2016-07-08 17:15:41",
    "description": "Burn loyalty points on Beyonce Song purchase."
  }
}

```

LoyaltyEarn Resource

The LoyaltyEarn resource credits the associated LoyaltyEarn.

IDs for loyalty earn transactions are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyEarn resource in JSON format.

```

{
  "id": "843G-838F-HY23-0238",
  "href": "loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/843G-838F-HY23-0238",
  "quantity": 20.00,
  "openingBalance": 280.00,
  "closingBalance": 300.00,
  "dateTime": "2016-07-29 12:18:51",
  "description": "A loyalty event triggered and points earned."
}

```

LoyaltyBurn Resource

The LoyaltyBurn resource credits the associated LoyaltyBurn.

IDs for loyalty earn transactions are strings that may consist of numbers and letters.

Below is a representation of the LoyaltyBurn resource in JSON format.

```
{
  "id": "834N-838F-3482-0238",
  "href": "loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/834N-838F-3482-0238",
  "quantity": 40.00,
  "openingBalance": 340.00,
  "closingBalance": 300.00,
  "dateTime": "2016-07-29 12:18:51",
  "description": "Burned loyalty points on album purchase."
}
```

LoyaltyEvent Resource

The LoyaltyEvent resource triggers the evaluation of a LoyaltyRule. Only LoyaltyRules with LoyaltyEventTypes matching the incoming event type is evaluated.

The LoyaltyEvent request format is that of the incoming event that triggers the loyalty event evaluation. Therefore, the only expected attribute of the LoyaltyEvent request is eventType.

```
{
  "eventType": "orderCreationNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "4343",
  "event": {
    "productOrder": {
      ...
    }
  }
}
```

LAYER RESOURCE

LoyaltyProgramProductSpec Resource

Field	Description
id	Unique identifier for the product specification.
name	The name of the product specification.
description	A narrative that explains in detail what the product specification is.
productNumber	An identification number assigned to uniquely identify the specification.
brand	The manufacturer or trademark of the specification.
needsLoyaltyAccount	If TRUE, a LoyaltyAccount is needed for each LoyaltyProgramProduct created according to the LoyaltyProgramProductSpec.
validFor	The period for which the product specification is valid.
lifeCycleStatus	The condition of the product specification, such as active, inactive, planned.
href	A reference to the product specification.

Figure 1 – LoyaltyProgramProductSpec resource model



Madene Marais 2016-8-15 10:50 AM

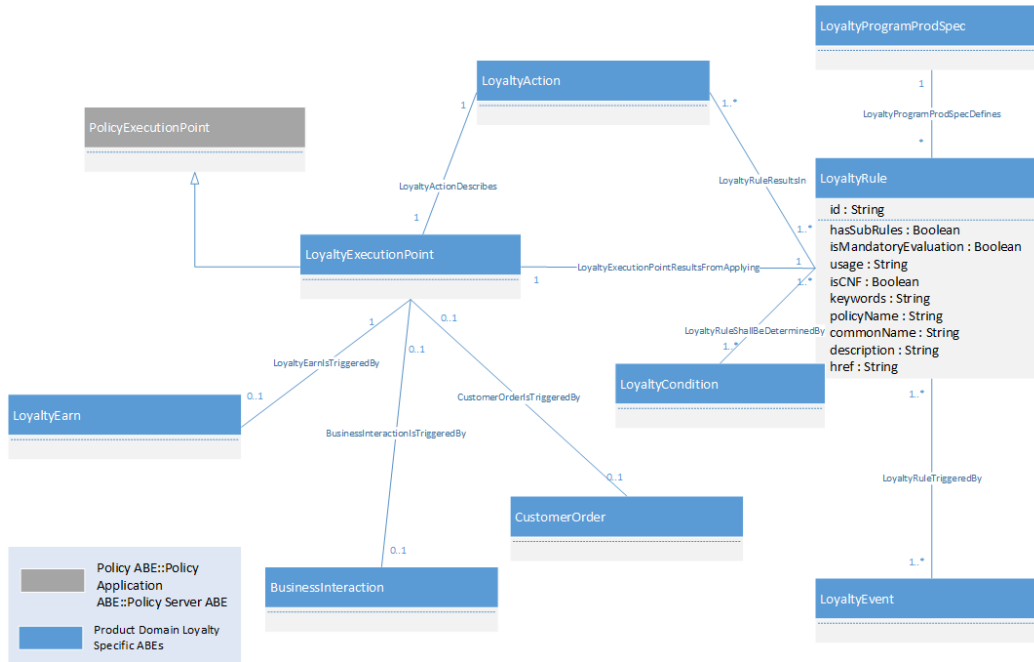
Comment [1]: Make other domains grey

LoyaltyRule Resource

Field	Description
id	Unique identifier for the loyalty rule.
isCNF	This is a Boolean attribute that, if TRUE, defines the condition clause of this rule to be represented in Conjunctive Normal Form (e.g., an AND of ORs). If the value of this attribute is FALSE, then the condition clause will be represented in Disjunctive Normal Form (e.g., an OR of ANDs).
hasSubRules	A Boolean attribute that signifies whether this loyalty rule has one or more sub-rules. Sub-rules are used to enforce a hierarchical nesting of rules, so that parent rules may control the execution and other semantics of sub-rules that they contain.
isMandatoryEvaluation	A Boolean attribute that, if TRUE, signifies that evaluation (and possibly action execution) of this entity is mandatory and must be attempted. If the Mandatory property value of this entity is FALSE, then the evaluation of this entity is considered to

	be "best effort" and may be ignored.
usage	A free-form string attribute that recommends how this policy object should be used.
keywords	A string attribute that defines a set of one or more keywords that a policy administrator may use to assist in characterizing or categorizing a policy object to facilitate search operations.
policyName	A generic naming attribute that can be used to identify different policy entities.
commonName	A user-friendly identifier of the loyalty rule.
description	A free-form description of the rule.
href	A reference to the loyalty rule.

Figure 2 – LoyaltyRule resource model



LoyaltyCondition Resource

Field	Description
id	Unique identifier for the loyalty condition.
attribute	Attribute to evaluate.
operator	Comparison operator to be used in the evaluation.
value	The value of the attribute to be evaluated.
href	Reference to the loyalty condition.

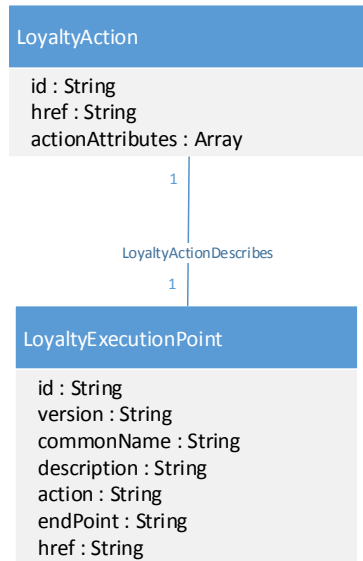
Figure 2 – LoyaltyCondition resource model

LoyaltyCondition
id : String
href : String
attribute : String
operator : String
value : String

LoyaltyAction Resource

Field	Description
id	Unique identifier for the loyalty action.
type	The type of loyalty action (LoyaltyEarn, CustomerOrder or BusinessInteraction).
actionAttributes	Additional attributes required to perform the action. These attributes will depend on the type of the loyalty action. For example, if it is a LoyaltyEarn action, there will be a “quantity” attribute value pair indicating how many points should be earned.
href	A reference to the loyalty action.
loyaltyExecutionPoint.version	A string that identifies the version of the loyalty execution point.
loyaltyExecutionPoint.commonName	A user-friendly identifier of the loyalty execution point.
loyaltyExecutionPoint.description	A free-form description of the loyalty execution point.
loyaltyExecutionPoint.action	The HTTP operation to be used when calling the endpoint (POST, PUT, GET or DELETE)
loyaltyExecutionPoint.endpoint	The endpoint to call to trigger a BusinessInteraction, CustomerOrder or LoyaltyEarn.

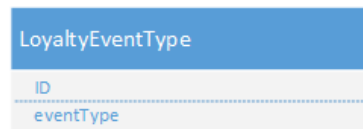
Figure 3 – LoyaltyAction resource model



LoyaltyEventType Resource

Field	Description
id	Unique identifier for the loyalty event type.
eventType	The type of loyalty event expected to trigger the loyalty rule evaluation, e.g. a CustomerOrder or an Invoice.

Figure 4 – LoyaltyEventType resource model

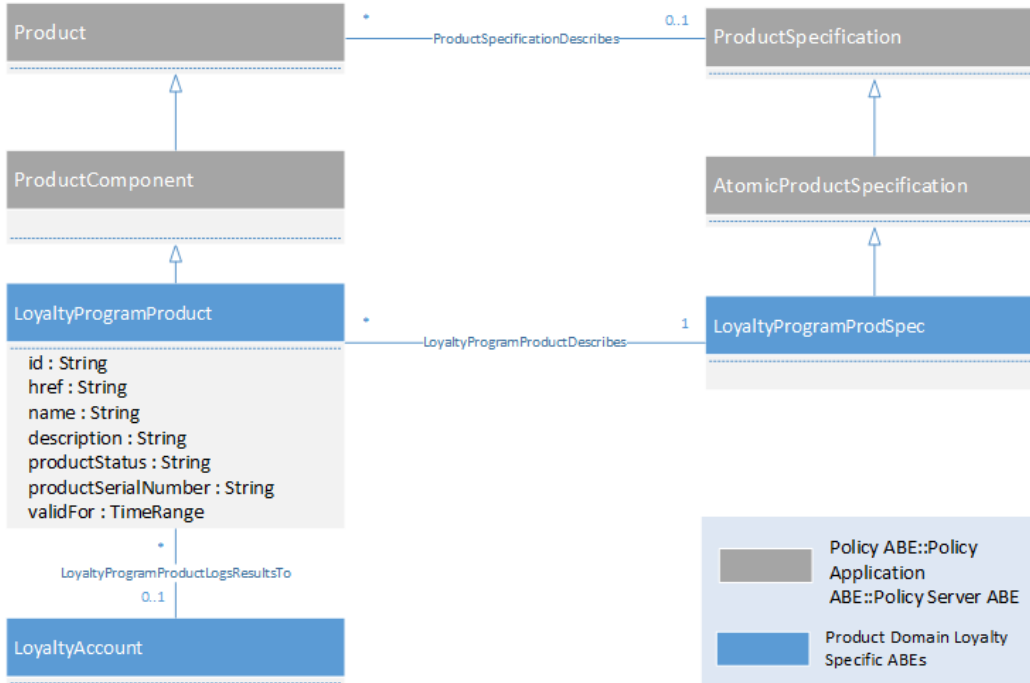


LoyaltyProgramProduct Resource

Field	Description
id	Unique identifier for the loyalty program product.
name	A word, term, or phrase by which the product is known and distinguished from other products.
description	An explanation of what the product is.
productStatus	The condition of the product, such as planned, designed, activated, disconnected.
productSerialNumber	A set of identifying characters and/or numbers assigned to, and usually

	marked on, each of a series of identical products.
validFor	The period during which the product is applicable.
loyaltyProgramProductSpec	The product specification that describes this product.
loyaltyAccount	The loyalty account to which results can be logged.
href	A reference to the loyalty program product.

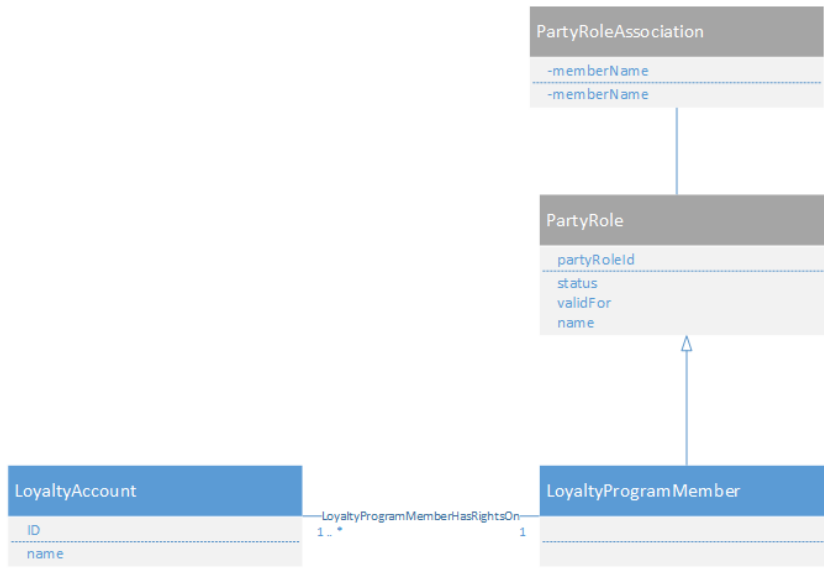
Figure 6 – LoyaltyProgramProduct resource model



LoyaltyProgramMember Resource

Field	Description
id	Unique identifier for the loyalty program member.
name	The loyalty member's name.
status	A free-form field in which the member's status may be captured, e.g. "active", "suspended", .etc.
validFor	The period during which the loyalty member is valid for.
loyaltyAccount	The loyalty accounts belonging to the loyalty member.
loyaltyProgramProduct	The loyalty products belonging to the loyalty member.
href	A reference to the loyalty program member.

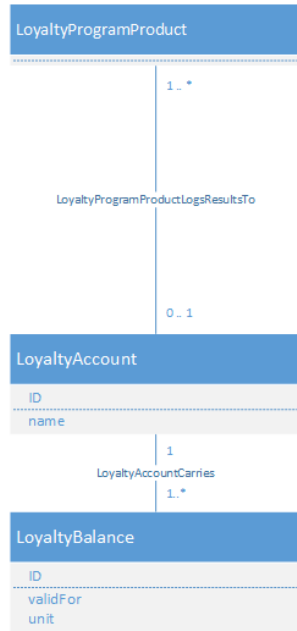
Figure 7 – LoyaltyProgramMember resource model



LoyaltyAccount Resource

Field	Description
id	Unique identifier for the loyalty account.
loyaltyProgramProduct	The loyalty program product associated with the loyalty account.
href	A reference to the loyalty member's loyalty account.

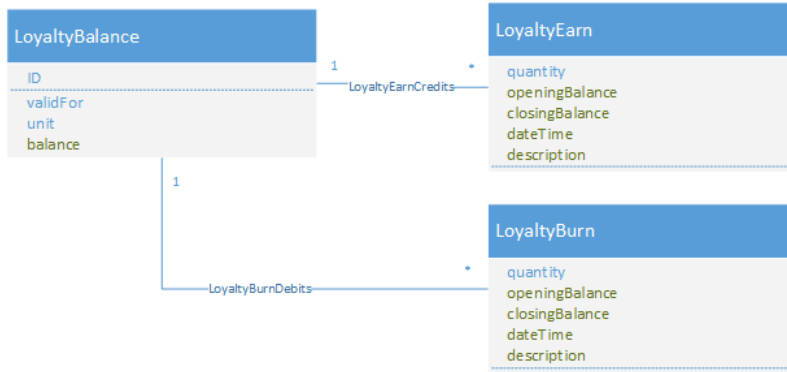
Figure 8 – LoyaltyAccount resource model



LoyaltyBalance Resource

Field	Description
id	Unique identifier for the loyalty balance.
unit	Unit of the quantity credited and debited from the balance.
balance	The current balance of the loyalty balance account.
validFor	The validity period in which loyalty can be burned using the account.
href	A reference to the loyalty member's loyalty balance.
loyaltyEarn	A collection of loyalty earned transactions on the balance account.
loyaltyBurn	A collection of loyalty burned transactions on the balance account.

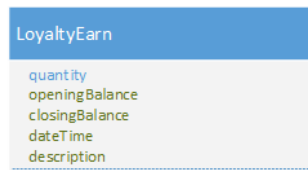
Figure 9 – LoyaltyBalance resource model



LoyaltyEarn Resource

Field	Description
id	Unique identifier for the loyalty earn transaction.
href	A reference to the loyalty earn transaction.
quantity	The amount of loyalty units earned.
openingBalance	The opening quantity on the account balance, before the loyalty units were earned.
closingBalance	The closing quantity on the account balance, after the loyalty units were earned.
dateTime	The date time on which the loyalty earn transaction occurred.
description	A free-form description describing the loyalty earn event.

Figure 10 – LoyaltyEarn resource model



LoyaltyBurn Resource

Field	Description
id	Unique identifier for the loyalty burn transaction.
href	A reference to the loyalty burn transaction.
quantity	The amount of loyalty units burned.
openingBalance	The opening quantity on the account balance, before the loyalty units were burned.
closingBalance	The closing quantity on the account balance, after the loyalty units were burned.

dateTime	The date time on which the loyalty burn transaction occurred.
description	A free-form description describing the loyalty burn event.

Figure 11 – LoyaltyBurn resource model

LoyaltyBurn
quantity
openingBalance
closingBalance
dateTime
description

LoyaltyEvent Resource

Field	Description
id	Unique identifier for the loyalty event.
eventType	The incoming loyalty event type.

Figure 12 – LoyaltyEvent resource model

LoyaltyEvent
ID
eventType

Event Models

loyaltyProgramMemberCreateNotification

```
{
  "eventType": "LoyaltyProgramMemberCreationNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "4343",
  "event": {
    "loyaltyProgramMember": {
      "status": "suspended",
      "name": "James Joe",
      "validFor": {
        "startDateTime": "2015-04-19T16:42:23.0Z ",
        "endDateTime": "2016-04-19T16:42:23.0Z"
      },
      "loyaltyAccount": [],
      "loyaltyProgramProduct": []
    }
  }
}
```

Field	Description
eventType	The event type. This value is LoyaltyProgramMemberCreateNotification for this event model.
eventId	A unique event identifier.
event.loyaltyProgramMember	The loyaltyProgramMember that was created.

loyaltyProgramMemberUpdateNotification

```
{
  "eventType": "LoyaltyProgramMemberUpdateNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "4343",
  "event": {
    "loyaltyProgramMember": {
      "status": "active",

```



```

"name": "James Joe",
"validFor": {
  "startDateTime": "2015-04-19T16:42:23.0Z ",
  "endDateTime": "2016-04-19T16:42:23.0Z"
},
"loyaltyAccounts": [{
  "id": "LoyaltyFund",
  "href": "/loyaltyManagement/loyaltyProgramMber/104/loyaltyAccount/LoyaltyFund",
  "loyaltyProgramProduct": {
    "id": "1211",
    "href":
"/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyProgramProduct/1211"
  },
  "loyaltyBalance": {
    "id": "iTunes",
    "balance": 1020,
    "validFor": {
      "startDateTime": "2015-04-19T16:42:23.0Z ",
      "endDateTime": "2016-04-19T16:42:23.0Z"
    },
    "unit": "magicPoints",
    "loyaltyEarn": {
      "quantity": 1000,
      "dateTime": "2016-04-19T16:42:23.0Z ",
      "openingBalance": 800,
      "closingBalance": 200
    }
  }
}],
"loyaltyProgramProduct": {
  "id": "1211",
  "name": "DataUsageBenefit",
  "description": "Data Usage Loyalty Benefits",
  "productStatus": "suspended",
  "productSerialNumber": "S2345666",
  "loyaltyProgramProductSpec": {
    "id": "121",
    "href": "/loyaltyManagement/loyaltyProgramProductSpec/121"
  }
}

```



```

    "href": "/loyaltyManagement/loyaltyProgramMber/104/loyaltyAccount/LoyaltyFund",
    "loyaltyProgramProduct": {
      "id": "1211",
      "href":
"/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyProgramProduct/1211"
    },
    "loyaltyBalance": [{
      "id": "iTunes",
      "balance": 1020,
      "validFor": {
        "startDateTime": "2015-04-19T16:42:23.0Z ",
        "endDateTime": "2016-04-19T16:42:23.0Z"
      },
      "unit": "magicPoints",
      "loyaltyEarn": {
        "quantity": 1000,
        "dateTime": "2016-04-19T16:42:23.0Z ",
        "openingBalance": 800,
        "closingBalance": 200
      }
    }
  ]
},
"loyaltyProgramProduct": {
  "id": "1211",
  "name": "DataUsageBenefit",
  "description": "Data Usage Loyalty Benefits",
  "productStatus": "suspended",
  "productSerialNumber": "S2345666",
  "loyaltyProgramProductSpec": {
    "id": "121",
    "href": "/loyaltyManagement/loyaltyProgramProductSpec/121"
  },
  "loyaltyAccount": {
    "id": "LoyaltyFund",
    "href": "/loyaltyManagement/loyaltyProgramMember/JDSU778DS/loyaltyAccount/LoyaltyFund"
  },
  "validFor": {
    "endDateTime": "2015-05-19T16:42:20Z",

```



```

      "startDateTime": "2017-05-19T16:42:20Z"
    },
    "href": "/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211"
  }
}
}

```

Field	Description
eventType	The event type. This value is <code>LoyaltyProgramMemberProductCreateNotification</code> for this event model.
eventId	A unique event identifier.
event.loyaltyProgramProduct	The <code>loyaltyProgramProduct</code> that was created.

loyaltyProgramMemberProductUpdateNotification

```

{
  "eventType": "LoyaltyProgramMemberProductUpdateNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "4343",
  "event": {
    "loyaltyProgramProduct": {
      "id": "1211",
      "name": "DataUsageBenefit",
      "description": "Data Usage Loyalty Benefits",
      "productStatus": "active",
      "productSerialNumber": "S2345666",
      "loyaltyProgramProductSpec": {
        "id": "121",
        "href": "/loyaltyManagement/loyaltyProgramProductSpec/121"
      },
    },
    "loyaltyAccount": {
      "id": "1",
      "href": "/loyaltyManagement/loyaltyProgramMember/121/loyaltyAccount/1"
    },
  },
  "validFor": {
    "endDateTime": "2017-05-19T16:42:20Z",
  }
}

```

```

      "startDateTime": "2017-05-19T16:42:20Z"
    },
    "href": "/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211"
  }
}
}

```

Field	Description
eventType	The event type. This value is <code>LoyaltyProgramMemberProductUpdateNotification</code> for this event model.
eventId	A unique event identifier.
event.loyaltyProgramProduct	The <code>loyaltyProgramProduct</code> that was updated.

loyaltyProgramMemberProductDeleteNotification

```

{
  "eventType": "LoyaltyProgramMemberProductDeleteNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "9438",
  "event": {
    "loyaltyProgramProduct": {
      "id": "1211",
      "name": "DataUsageBenefit",
      "description": "Data Usage Loyalty Benefits",
      "productStatus": "active",
      "productSerialNumber": "S2345666",
      "loyaltyProgramProductSpec": {
        "id": "121",
        "href": "/loyaltyManagement/loyaltyProgramProductSpec/121"
      },
    },
    "loyaltyAccount": {
      "id": "1",
      "href": "/loyaltyManagement/loyaltyProgramMember/121/loyaltyAccount/1"
    },
  },
  "validFor": {
    "endDateTime": "2017-05-19T16:42:20Z",
  }
}

```

```

    "startDateTime": "2017-05-19T16:42:20Z"
  },
  "href": "/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211"
}
}
}

```

Field	Description
eventType	The event type. This value is <code>LoyaltyProgramMemberProductDeleteNotification</code> for this event model.
eventId	A unique event identifier.
event.loyaltyProgramProduct	The <code>loyaltyProgramProduct</code> before it was deleted.

loyaltyEarnNotification

```

{
  "eventType": "LoyaltyEarnNotification",
  "eventTime": "2016-06-04T07:42:25.0Z",
  "eventId": "3983",
  "event": {
    "loyaltyEarn": {
      "id": "843G-838F-HY23-0238",
      "href": "loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/
iTunes/loyaltyEarn/843G-838F-HY23-0238",
      "quantity": 20.00,
      "openingBalance": 280.00,
      "closingBalance": 300.00,
      "dateTime": "2016-07-29 12:18:51",
      "description": "A loyalty event triggered and points earned."
    }
  }
}
}

```

Field	Description
eventType	The event type. This value is LoyaltyEarnNotification for this event model.
eventId	A unique event identifier.
event.loyaltyEarn	The loyaltyEarn that was transacted.

loyaltyBurnNotification

```
{
  "eventType": "LoyaltyBurnNotification",
  "eventTime": "2015-02-04T07:42:25.0Z",
  "eventId": "3435",
  "event": {
    "loyaltyBurn": {
      "id": "843G-838F-HY23-0238",
      "href": "loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/843G-838F-HY23-0238",
      "quantity": 20.00,
      "openingBalance": 280.00,
      "closingBalance": 260.00,
      "dateTime": "2015-02-04 07:42:20",
      "description": "Burn loyalty points on Beyonce Song purchase."
    }
  }
}
```

Field	Description
eventType	The event type. This value is LoyaltyBurnNotification for this event model.
eventId	A unique event identifier.
event.loyaltyBurn	The loyaltyBurn that was transacted.

loyaltyEventNotification

```
{
  "eventType": "LoyaltyEarnNotification",
  "eventTime": "2016-06-04T07:42:25.0Z",
  "eventId": "3983",
```



```
"event": {
  "loyaltyEvent": {
    All the incoming event attributes.
  }
}
```

Field	Description
eventType	The event type. This value is LoyaltyEvent for this event model.
eventId	A unique event identifier.
event.loyaltyEvent	All the attributes incoming from the LoyaltyEvent.

API OPERATION TEMPLATES

GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyProgramProductSpec with no {ID}.

Description :

- This operation is used to retrieve the loyalty program product specification. A loyalty program product specification contains loyalty rules.
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty program product specifications matched supplied criteria)
- 404 Not Found – The loyalty program product specification could not be found. This return code is only applicable when a managed entity is requested.

REQUEST
GET /loyaltyManagement/loyaltyProgramProductSpec/121 Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>{ "id": "121", "name": "UpComingProfessionalsProgram", "description": "Loyalty Program to ensure that prepaid youth market is retained", "productNumber": "983284", "brand": "Globetom", "needsLoyaltyAccount": true, "lifeCycleStatus": "suspended", "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121", "validFor": {</pre>

```
"endTime": "2016-12-31T23:59:59Z",
"startTime": "2016-01-01T00:00:00Z"
},
"loyaltyRule": [{
  "id": "1",
  "name": "YouthRule",
  "description": "Verify if the customers age qualifies for youth program benefits",
  "isCNF": true,
  "hasSubRules": false,
  "isMandatoryEvaluation": true,
  "usage": "Subscribers younger than 23.",
  "keywords": "age,youth",
  "policyName": "Age less than 23",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1",
  "loyaltyEventType": [{
    "id": "3",
    "eventType": "orderCreationNotification",
    "href": "http://server:port/loyaltyManagement/loyaltyEventType/3"
  ]},
  "loyaltyCondition": [{
    "id": "1",
    "attribute": "age",
    "operator": "<",
    "value": "23",
    "href": "http://server:port/loyaltyManagement/loyaltyCondition/1"
  ]},
  "loyaltyAction": [{
    "id": "111",
    "type": "LoyaltyEarn",
    "actionAttributes": {
      "quantity": 50
    },
    "href": "/loyaltyManagement/loyaltyAction/111"
  }],
  "loyaltyExecutionPoint": {
    "commonName": "Earn50",
    "description": "",
    "action": "POST",
```

```

    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
                loyaltyAccount/{accountId}/loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "2.0"
  },
}
}

```

PUT /loyaltyManagement/loyaltyProgramProductSpec/{ID}

PUT is not supported for loyaltyProgramProductSpec since the loyaltyRule resource cannot be updated on this endpoint. Any modification can be handled through the PATCH API.

PATCH /loyaltyManagement/loyaltyProgramProductSpec/{ID}

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- This operation partially updates a loyalty program product specification.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty program product specification to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Patchable attributes are shown in the table below.

Attribute name	Patchable	Rule
id	N	
name	Y	
description	Y	
productNumber	Y	

brand	Y	
needsLoyaltyAccount	N	
validFor	Y	
lifeCycleStatus	Y	
loyaltyRule	N	The loyaltyRule can only be updated on its own route.

REQUEST

PATCH /loyaltyManagement/loyaltyProgramProductSpec/121

Content-Type: application/json

```
{
  "name": "UpComingProfessionalsProgram",
  "description": "Loyalty Program to ensure that prepaid youth market is retained",
}
```

RESPONSE

201

Content-Type: application/json

```
{
  "name": "UpComingProfessionalsProgram",
  "description": "Loyalty Program to ensure that prepaid youth market is retained",
  "productNumber": "983284",
  "brand": "Globetom",
  "needsLoyaltyAccount": true,
  "lifeCycleStatus": "suspended",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121",
  "validFor": {
    "endDateTime": "2016-12-31T23:59:59Z",
    "startDateTime": "2016-01-01T00:00:00Z"
  },
  "loyaltyRule": []
}
```

POST /loyaltyManagement/loyaltyProgramProductSpec

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation is used to create a new loyalty program product specification.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty program product specification was created
- 409 Conflict – Resource already exists.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Attribute name	Mandatory	Default	Rule
id	N	Automatically generated	If not given, the id is generated by the system. It is also possible to add an ID in the POST request, which is then used
name	Y		
description	N		
productNumber	Y		
brand	N		
needsLoyaltyAccount	N	false	
validFor	N		
lifeCycleStatus	N	active	

REQUEST

```

POST /loyaltyManagement/loyaltyProgramProductSpec
{
  "id" : "121",
  "name":"UpComingProfessionalsProgram",
  "productNumber":"121",
  "description": "Loyalty Program to ensure that prepaid youth market is retained",
  "needsLoyaltyAccount": true,
  "lifeCycleStatus": "active",
  "brand" : "Globetom",
  "validFor" : {
    "startDateTime" : "2016-01-01T00:00:00Z",
    "endDateTime" : "2016-12-31T23:59:59Z"
  }
}

```

RESPONSE

```

201
Content-Type: application/json
Content-Location: loyaltyManagement/loyaltyProgramProductSpec/121
{
  "id" : "121",
  "name":"UpComingProfessionalsProgram",
  "productNumber":"121",
  "description": "Loyalty Program to ensure that prepaid youth market is retained",
  "needsLoyaltyAccount": true,
  "lifeCycleStatus": "active",
  "brand" : "Globetom",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121"
  "validFor" : {
    "startDateTime" : "2016-01-01T00:00:00Z",
    "endDateTime" : "2016-12-31T23:59:59Z"
  },
  "loyaltyRule": []
}

```

DELETE /LOYALTYMANAGEMENT/LOYALTYPROGRAMPRODUCTSPEC/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to delete a loyalty program product specification.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty program product specification was deleted.
- 404 Not Found – The loyalty program product specification could not be found.

REQUEST
DELETE /loyaltyManagement/loyaltyProgramProductSpec/78 Accept: application/json
RESPONSE
200 Content-Type: application/json

GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule with no {ID}.

Description :

- This operation is used to retrieve the loyalty program product specification's loyalty rule(s).
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty rules matched supplied criteria)
- 404 Not Found – The loyalty rule could not be found. This return code is only applicable when a managed entity is requested.

REQUEST

GET /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1
 Accept: application/json

RESPONSE

```
200
Content-Type: application/json
{
  "id": "1",
  "commonName": "",
  "description": "Verify if the customers age qualifies for youth program benefits",
  "isCNF": true,
  "hasSubRules": false,
  "isMandatoryEvaluation": true,
  "usage": "Subscribers younger than 23.",
  "keywords": "age,youth",
  "policyName": "Age less than 23",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121/LoyaltyRule/1",
  "loyaltyEventType": [],
  "loyaltyCondition": [],
  "loyaltyAction": []
}
```

PUT /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}

Description

- Update the complete loyalty rule.
- The resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty rule to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

REQUEST

PUT /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/222

```
Content-Type: application/json
{
  "description": "Verify if the loyalty member status is active",
  "isCNF": true,
  "hasSubRules": false,
  "isMandatoryEvaluation": true,
  "usage": "Active subscribers",
  "keywords": "status,active",
  "policyName": "Active subscribers"
}
```

RESPONSE

```
201
Content-Type: application/json
{
  "id": "222",
  "description": "Verify if the loyalty member status is active",
  "isCNF": true,
  "hasSubRules": false,
  "isMandatoryEvaluation": true,
  "usage": "Active subscribers",
  "keywords": "status,active",
  "policyName": "Active subscribers",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/222",
  "loyaltyEventType": [],
  "loyaltyCondition": [],
  "loyaltyAction": []
}
```

PATCH /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}

Description :

- This operation partially updates a loyalty rule.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty rule to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

Patchable attributes are shown in the table below.

Attribute name	Patchable	Rule
id	N	
isCNF	Y	
hasSubRules	Y	
isMandatoryEvaluation	Y	
usage	Y	
keywords	Y	
policyName	Y	
commonName	Y	
description	Y	
loyaltyEventType	N	The loyaltyEventType is updated on its own route.
loyaltyCondition	N	The loyaltyCondition is updated on its own route.
loyaltyAction	N	The loyaltyAction is updated on its own route.

REQUEST

```
PATCH /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/222
Content-Type: application/json
{
```

```

"isCNF": false,
"isMandatoryEvaluation" : false,
}

```

RESPONSE

```

200
Content-Type: application/json
{
  "id": "222",
  "description": "Verify if the loyalty member status is active",
  "isCNF": false,
  "hasSubRules" : false,
  "isMandatoryEvaluation" : false,
  "usage" : "Active subscribers",
  "keywords" : "status,active",
  "policyName" : "Active subscribers"
  "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/222",
  "loyaltyEventType": [],
  "loyaltyCondition": [],
  "loyaltyAction": []
}

```

POST /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule

Description :

- This operation is used to create a new loyalty rule.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty rule was created
- 409 Conflict – Resource already exists.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Attribute name	Mandatory	Default	Rule
id	N	Automatically generated	If not given, the id is generated by the system.

			It is also possible to add an ID in the POST request, which is then used
isCNF	N	true	
hasSubRules	N	true	
isMandatoryEvaluation	N	true	
usage	N		
keywords	N		
policyName	N		
commonName	N		
description	N		

REQUEST

POST /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule

Content-Type: application/json

```
{
  "id" : "1",
  "description": "Verify if the customers age qualifies for youth program benefits",
  "isCNF": true,
  "hasSubRules" : false,
  "isMandatoryEvaluation" : true,
  "usage" : "Subscribers younger than specified age.",
  "keywords" : "age,youth",
  "policyName" : "Age less than 23"
}
```

RESPONSE

201

Content-Type: application/json

```

Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1
{
  "id" : "1",
  "description": "Verify if the customers age qualifies for youth program benefits",
  "isCNF": true,
  "hasSubRules" : false,
  "isMandatoryEvaluation" : true,
  "usage" : "Subscribers younger than specified age.",
  "keywords" : "age,youth",
  "policyName" : "Age less than 23",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1",
  "loyaltyEventType": [],
  "loyaltyCondition": [],
  "loyaltyAction": []
}

```

DELETE /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to delete a loyalty rule.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty rule was deleted.
- 404 Not Found – The loyalty rule could not be found.

REQUEST

```

DELETE /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1
Accept: application/json

```

RESPONSE

```

200
Content-Type: application/json

```

GET

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition with no {ID}.

Description :

- This operation is used to retrieve the loyalty rule's loyalty condition(s).
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty conditions matched supplied criteria)
- 404 Not Found – The loyalty rule condition could not be found. This return code is only applicable when a managed entity is requested.

REQUEST

```
GET /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyCondition/2
Accept: application/json
```

RESPONSE

200

Content-Type: application/json

```
{
  "id": "2",
  "attribute": "age",
  "operator": "<",
  "value": "30",
  "href": "http://server:port/loyaltyManagement/loyaltyCondition/2"
}
```

PUT

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID}

PUT is not supported for loyaltyCondition under loyaltyRule since a condition can either be added to a loyalty rule using the POST API or removed from a loyalty rule using the DELETE API.

PATCH

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID}

PATCH is not supported for loyaltyCondition under loyaltyRule since a condition can either be added to a loyalty rule using the POST API or removed from a loyalty rule using the DELETE API.

POST

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID}

Description :

- This operation is used to add a loyalty condition to a loyalty rule.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty condition was added to the rule.
- 409 Conflict – The loyalty condition has already been added to the loyalty rule.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute, for example the loyaltyCondition does not exist.

Attribute name	Mandatory	Default	Rule
id	Y		It should be an existing loyaltyCondition

REQUEST

POST /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyCondition

Content-Type: application/json

```
{
```


<pre>"id" : "2" }</pre>
RESPONSE
<pre>201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyCondition/2 { "id": "2", "attribute": "age", "operator": "<", "value": "30", "href": "http://server:port/loyaltyManagement/loyaltyCondition/2" }</pre>

DELETE

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyCondition/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to remove a loyalty condition from a loyalty rule.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty condition was removed from the loyalty rule.
- 404 Not Found – The loyalty condition could not be found.

REQUEST
<pre>DELETE /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyCondition/2 Accept: application/json</pre>
RESPONSE
<pre>200</pre>

Content-Type: application/json

GET
**/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEvent
 Type/{ID}**

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET
 /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventType with no {ID}.

Description :

- This operation is used to retrieve the loyalty rule's supported loyalty event(s).
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty event types matched supplied criteria)
- 404 Not Found – The loyalty event type could not be found. This return code is only applicable when a managed entity is requested.

REQUEST
GET /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyEventType/2 Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>{ "id": "2", "eventType": "orderCreationNotification", "href": "http://server:port/loyaltyManagement/loyaltyEventType/2" }</pre>

PUT

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventtype/{ID}

PUT is not supported for loyaltyEventtype under loyaltyRule since an event type can either be added to a loyalty rule using the POST API or removed from a loyalty rule using the DELETE API.

PATCH

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventtype/{ID}

PATCH is not supported for loyaltyEventtype under loyaltyRule since an event type can either be added to a loyalty rule using the POST API or removed from a loyalty rule using the DELETE API.

POST

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEventtype

Description :

- This operation is used to add a loyalty event type to a loyalty rule.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty event type was added to the loyalty rule
- 409 Conflict – The loyalty event type has already been added to the loyalty rule.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute, for example the loyaltyEventtype does not exist.

Attribute name	Mandatory	Default	Rule
id	Y		It should be an existing loyaltyEventtype

REQUEST

```
POST /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyEventtype
Content-Type: application/json
{
```

<pre>"id" : "2" }</pre>
RESPONSE
<pre>201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyEventType/2 { "id": "2", "eventType": "orderCreationNotification", "href": "http://server:port/loyaltyManagement/loyaltyEventType/2" }</pre>

DELETE

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyEvent
 ventType/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to remove a loyalty event type from a loyalty rule.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty event type was removed from the loyalty rule.
- 404 Not Found – The loyalty event type could not be found.

REQUEST
<pre>DELETE /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/2/loyaltyEventType/1 Accept: application/json</pre>
RESPONSE
<pre>200 Content-Type: application/json</pre>

GET

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET `/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction` with no `{ID}`.

Description :

- This operation is used to retrieve the loyalty rule's loyalty action(s).
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty actions matched supplied criteria)
- 404 Not Found – The loyalty action could not be found. This return code is only applicable when a managed entity is requested.

REQUEST

```
GET /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyAction/2
Accept: application/json
```

RESPONSE

```
200
Content-Type: application/json
{
  "id": "2",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/2",
  "loyaltyExecutionPoint": {
    "commonName": "Earn50",
    "description": ""
  }
}
```

```

    "action": "POST",
    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "2.0"
  }
}

```

PUT

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID}

PUT is not supported for loyaltyAction under loyaltyRule since an action can either be added to a loyalty rule using the POST API or removed from a loyalty rule using the DELETE API.

PATCH

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID}

PATCH is not supported for loyaltyAction under loyaltyRule since an action can either be added to a loyalty rule using the POST API or removed from a loyalty rule using the DELETE API.

POST

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction

Description :

- This operation is used to add a loyalty action to a loyalty rule.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty action was added to the loyalty rule.
- 409 Conflict – The loyalty action has already been added to the loyalty rule.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute, for example the loyalty action does not exist.

Attribute name	Mandatory	Default	Rule
id	Y		It should be an existing

			loyaltyAction
--	--	--	---------------

REQUEST

POST /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyAction

Content-Type: application/json

```
{
  "id" : "2"
}
```

RESPONSE

201

Content-Type: application/json

Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyAction/2

```
{
  "id": "2",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/2",
  "loyaltyExecutionPoint": {
    "commonName": "Earn50",
    "description": "",
    "action": "POST",
    "endpoint":
      "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "2.0"
  }
}
```

DELETE

/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyAction/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to remove a loyalty action from a loyalty rule.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty action was removed from the loyalty rule.
- 404 Not Found – The loyalty action could not be found.

REQUEST

```
DELETE /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyAction/1
Accept: application/json
```

RESPONSE

```
200
Content-Type: application/json
```

GET /loyaltyManagement/loyaltyCondition/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyCondition with no {ID}.

Description :

- This operation is used to retrieve the loyalty condition.
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty conditions matched supplied criteria)
- 404 Not Found – The loyalty condition could not be found. This return code is only applicable when a managed entity is requested.

REQUEST

GET /loyaltyManagement/loyaltyCondition/1
Accept: application/json

RESPONSE

```
200
Content-Type: application/json
{
  "id": "1",
  "attribute": "age",
  "operator": "<",
  "value": "30",
  "href": "http://server:port/loyaltyManagement/loyaltyCondition/1"
}
```

PUT /loyaltyManagement/loyaltyCondition/{ID}

Description

- Update the complete loyalty condition.
- The resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty condition to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

REQUEST

```
PUT /loyaltyManagement/loyaltyCondition/1
Content-Type: application/json
{
  "attribute": "status",
  "operator": "=",
  "value": "active"
}
```

RESPONSE

```

201
Content-Type: application/json
{
  "id" : "2",
  "attribute": "status",
  "operator": "=",
  "value": "active",
  "href": "http://server:port/loyaltyManagement/loyaltyCondition/1"
}

```

PATCH /loyaltyManagement/loyaltyCondition/{ID}

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- This operation partially updates a loyalty condition.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty condition to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

Patchable attributes are shown in the table below.

Attribute name	Patchable	Rule
id	N	
attribute	Y	
operator	Y	
value	Y	

REQUEST

```
PUT /loyaltyManagement/loyaltyCondition/1
```

```
Content-Type: application/json
```

```
{
  "operator": "<=",
}
```

RESPONSE

```
200
```

```
Content-Type: application/json
```

```
{
  "id": "1",
  "attribute": "age",
  "operator": "<=",
  "value": "30",
  "href": "http://server:port/loyaltyManagement/loyaltyCondition/1"
}
```

POST /loyaltyManagement/loyaltyCondition

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation is used to create a new loyalty condition.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty condition was created
- 409 Conflict – Resource already exists.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Attribute name	Mandatory	Default	Rule
id	N	Automatically generated	If not given, the id is generated by the system. It is also possible to add an ID in

			the POST request, which is then used
attribute	Y		
operator	Y		
value	Y		

REQUEST

POST /loyaltyManagement/loyaltyCondition

```
{
  "id": "2",
  "attribute": "status",
  "operator": "=",
  "value": "active"
}
```

RESPONSE

201

Content-Type: application/json

Content-Location: loyaltyManagement/loyaltyCondition/2

```
{
  "id": "2",
  "attribute": "status",
  "operator": "=",
  "value": "active",
  "href": "http://server:port/loyaltyManagement/loyaltyCondition/2"
}
```

DELETE /loyaltyManagement/loyaltyCondition/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to delete a loyalty condition.
- This resource represents a managed entity.

- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty condition was deleted.
- 404 Not Found – The loyalty condition could not be found.
- 422 Unprocessable Entity –The loyalty condition belongs to a loyalty rule and cannot be deleted.

REQUEST
DELETE /loyaltyManagement/loyaltyProgramCondition/2 Accept: application/json
RESPONSE
200 Content-Type: application/json

GET /loyaltyManagement/loyaltyAction/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyAction with no {ID}.

Description :

- This operation is used to retrieve the loyalty action. A loyalty action contains a loyalty execution point.
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty actions matched supplied criteria)
- 404 Not Found – The loyalty action could not be found. This return code is only applicable when a managed entity is requested.

REQUEST
GET /loyaltyManagement/loyaltyAction/111 Accept: application/json

RESPONSE

200

Content-Type: application/json

```
{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/111",
  "loyaltyExecutionPoint": {
    "commonName": "Earn50",
    "description": "",
    "action": "POST",
    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
      loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "2.0"
  }
}
```

PUT /loyaltyManagement/loyaltyAction/{ID}

Description

- Update the complete loyalty action.
- The resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty action to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

REQUEST

PUT /loyaltyManagement/loyaltyAction/2

```
Content-Type: application/json
{
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 80
  },
  "loyaltyExecutionPoint": {
    "commonName": "Earn80",
    "description": "Award customer loyalty",
    "action": "POST",
    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
                loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "5.0"
  }
}
```

RESPONSE

```
201
Content-Type: application/json
{
  "id": "2",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 80
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/2",
  "loyaltyExecutionPoint": {
    "commonName": "Earn80",
    "description": "Award customer loyalty",
    "action": "POST",
    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
                /loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "5.0"
  }
}
```

PATCH /loyaltyManagement/loyaltyAction/{ID}

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- This operation partially updates a loyalty action.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty action to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

Patchable attributes are shown in the table below.

Attribute name	Patchable	Rule
id	N	
type	Y	
actionAttributes	Y	
loyaltyExecutionPoint.version	Y	Should be higher than the previous version.
loyaltyExecutionPoint.commonName	Y	
loyaltyExecutionPoint.description	Y	
loyaltyExecutionPoint.action	Y	
loyaltyExecutionPoint.endpoint	Y	

REQUEST

```
PATCH /loyaltyManagement/loyaltyAction/2
```


Content-Type: application/json

```
{
  "actionAttributes": {
    "quantity": 100
  },
  "loyaltyExecutionPoint": {
    "version": "6.0"
  }
}
```

RESPONSE

201

Content-Type: application/json

```
{
  "id": "2",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 100
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/2",
  "loyaltyExecutionPoint": {
    "commonName": "Earn100",
    "description": "Award customer loyalty",
    "action": "POST",
    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "6.0"
  }
}
```

POST /loyaltyManagement/loyaltyAction

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation is used to create a new loyalty action.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty action was created
- 409 Conflict – Resource already exists.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Attribute name	Mandatory	Default	Rule
id	N	Automatically generated	If not given, the id is generated by the system. It is also possible to add an ID in the POST request, which is then used
type	Y		Must be LoyaltyEarn, CustomerOrder or BusinessInteraction
actionAttributes	Y		
loyaltyExecutionPoint.version	N	1.0	
loyaltyExecutionPoint.commonName	N		
loyaltyExecutionPoint.description	N		
loyaltyExecutionPoint.action	Y		POST, PUT, GET or DELETE
loyaltyExecutionPoint.endpoint	Y		

REQUEST

POST /loyaltyManagement/loyaltyAction

```
{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
}
```

```

"loyaltyExecutionPoint": {
  "commonName": "LoyaltyEarn",
  "description": "",
  "action": "POST",
  "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
              /loyaltyBalance/{balanceId}/loyaltyEarn",
  "version": "2.0"
}
}

```

RESPONSE

```

201
Content-Type: application/json
Content-Location: loyaltyManagement/loyaltyAction/111
{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/111",
  "loyaltyExecutionPoint": {
    "commonName": "LoyaltyEarn",
    "description": "",
    "action": "POST",
    "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
                loyaltyBalance/{balanceId}/loyaltyEarn",
    "version": "2.0"
  }
}

```

DELETE /loyaltyManagement/loyaltyAction/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to delete a loyalty action.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty action was deleted.
- 404 Not Found – The loyalty action could not be found.
- 422 Unprocessable Entity –The loyalty action belongs to a loyalty rule and cannot be deleted.

REQUEST
DELETE /loyaltyManagement/loyaltyAction/78 Accept: application/json
RESPONSE
200 Content-Type: application/json

GET /loyaltyManagement/loyaltyEventType/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyEventType with no {ID}.

Description :

- This operation is used to retrieve the loyalty event types.
- Filtering selection is enabled on all attributes.
- Attribute selection is enabled on all attributes.
- The resource is either a managed entity or a collection, depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty event types matched supplied criteria)
- 404 Not Found – The loyalty event type could not be found. This return code is only applicable when a managed entity is requested.

REQUEST
GET /loyaltyManagement/loyaltyEventType/111 Accept: application/json
RESPONSE

```

200
Content-Type: application/json
{
  "id": "111",
  "eventType": "orderCreationNotification",
  "href": "http://server:port/loyaltyManagement/loyaltyEventType/111"
}

```

PUT /loyaltyManagement/loyaltyEventType/{ID}

Description

- Update the complete loyalty event type.
- The resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty event type to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

REQUEST

```

PUT /loyaltyManagement/loyaltyEventType/2
Content-Type: application/json
{
  "eventType": "createCustomerInvoice"
}

```

RESPONSE

```

201
Content-Type: application/json
{
  "id": "2",
  "eventType": "createCustomerInvoice",
  "href": "http://server:port/loyaltyManagement/loyaltyEventType/2"
}

```

```
}

```

PATCH /loyaltyManagement/loyaltyEventType/{ID}

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- This operation partially updates a loyalty event type.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty event type to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

Patchable attributes are shown in the table below.

Attribute name	Patchable	Rule
id	N	
eventType	Y	

REQUEST

```
PATCH /loyaltyManagement/loyaltyEventType/2
Content-Type: application/json
{
  "eventType": "createCustomerInvoice"
}
```

RESPONSE

```
201
Content-Type: application/json
{
```

```

{id": "2",
"eventType": "createCustomerInvoice",
"href": "http://server:port/loyaltyManagement/loyaltyEventType/111"
}

```

POST /loyaltyManagement/loyaltyEventType

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation is used to create a new loyalty event type.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty event type was created
- 409 Conflict – Resource already exists.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Attribute name	Mandatory	Default	Rule
id	N	Automatically generated	If not given, the id is generated by the system. It is also possible to add an ID in the POST request, which is then used
eventType	Y		Corresponds to the eventType attribute of external systems registered to trigger a LoyaltyEvent.

REQUEST

```

POST /loyaltyManagement/loyaltyEventType
{
  "id": "111",
  "eventType": "orderCreationNotification"
}

```

}
RESPONSE
201 Content-Type: application/json Content-Location: loyaltyManagement/loyaltyEventType/111 { "id": "111", "eventType": "orderCreationNotification", "href": "http://server:port/loyaltyManagement/loyaltyEventType/111" }

DELETE /loyaltyManagement/loyaltyEventType/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to delete a loyalty event type.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty event type was deleted.
- 404 Not Found – The loyalty event type could not be found.
- 422 Unprocessable Entity –The loyalty event type belongs to a loyalty rule and cannot be deleted.

REQUEST
DELETE /loyaltyManagement/loyaltyEventType/78 Accept: application/json
RESPONSE
200 Content-Type: application/json

GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or collection.

Note that collections can be retrieved via GET
/loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct with no {ID}.

Description :

- This operation is used to retrieve the loyalty program member's loyalty program product(s).
- Filtering selection is enabled on all first level attributes but not on inner classes.
- Attribute selection is enabled on all first level attributes but not on inner classes.
- The resource is either a managed entity or a collection depending on the query pattern.
- The ID may be a string (or a string containing numbers).

Behavior :

Returned status codes:

- 200 OK - The request was successful (includes situation in which no loyalty program products matched supplied criteria)
- 404 Not Found – The loyalty program product could not be found. This return code is only applicable when a managed entity is requested.

REQUEST
GET /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211 Accept: application/json
RESPONSE
200 Content-Type: application/json { "id": "1211", "name": "DataUsageBenefit", "description": "Data Usage Loyalty Benefits", "productStatus": "suspended", "productSerialNumber": "S2345666", "loyaltyProgramProductSpec": { "id": "121", "href" : "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121" }, "loyaltyAccount": { "id": "1", "href" : "http://server:port/loyaltyManagement/loyaltyProgramMember/121/loyaltyAccount/1" }, }

```

"validFor": {
  "endDateTime": "2017-05-19T16:42:20Z",
  "startDateTime": "2017-05-19T16:42:20Z"
},
"href": "http://server:port/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211"
}

```

PUT /loyaltyManagement/ loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}

PUT is not supported for LoyaltyProgramProduct as any modification can be handled the through PATCH API.

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- This operation partially updates a loyalty program member's loyalty program product.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Return status codes

- 201 OK – The update was successful.
- 404 Not Found – The loyalty program product to be updated could not be found.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute

Patchable attributes are shown in the table below.

Attribute name	Patchable	Rule
id	N	
name	Y	
description	Y	

productStatus	Y	
productSerialNumber	Y	
validFor	Y	
productSpecId	N	
accountId	N	

REQUEST

PATCH /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211

Content-Type: application/json

```
{
  "description": "Data Usage Loyalty Benefits",
  "productStatus": "active"
}
```

RESPONSE

201

Content-Type: application/json

```
{
  "id": "1211",
  "name": "DataUsageBenefit",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211",
  "description": "Data Usage Loyalty Benefits",
  "productStatus": "active",
  "productSerialNumber": "S2345666",
  "validFor": {
    "endTime": "2017-05-19T16:42:20Z",
    "startTime": "2016-04-19T12:00:00Z"
  }
}
```

POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation is used to create a new loyalty program product.
- This resource represents a managed object entity.

Behavior :

Return status codes

- 201 Created – The loyalty program product was created.
- 409 Conflict – Resource already exists.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Attribute name	Mandatory	Default	Rule
id	N	Automatically generated	If not given, the id is generated by the system. It is also possible to add an ID in the POST request, which is then used
name	Y		
description	N		
productStatus	N	activated	
productSerialNumber	Y		
validFor	N		
productSpecId	Y		Should be an existing LoyaltyProgramProductSpecification
accountId	N		Should be an existing LoaltyAccount. If the LoyaltyProgramProductSpec's needsLoyaltyAccount attribute is TRUE and the accountId attribute was omitted, then a new loyaltyAccount will be added to the loyaltyProgramMember. If the accountId attribute was provided and the LoyaltyProgramProductSpec's needsLoyaltyAccount attribute is FALSE, an error needs to be

			<p>returned.</p> <p>If the accountId attribute was provided and the LoyaltyProgramProductSpec's needsLoyaltyAccount attribute is TRUE, it has to correspond to one of the LoyaltyMember's existing loyaltyAccounts.</p>
--	--	--	---

REQUEST

POST /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct

Content-Type: application/json

```
{
  "id": "1213",
  "name": "PrepaidTopupBenefits",
  "description": "Prepaid account top-up benefits",
  "productStatus": "active",
  "productSerialNumber": "S23458",
  "validFor": {
    "endDateTime": "2017-05-19T16:42:20Z",
    "startDateTime": "2016-04-19T12:00:00Z"
  },
  "productSpecId": "1",
  "accountId": "123"
}
```

RESPONSE

201

Content-Type: application/json

Content-Location: /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1213

```
{
  "id": "1213",
  "name": "PrepaidTopupBenefits",
  "description": "Prepaid account top-up benefits",
  "productStatus": "active",
  "productSerialNumber": "S23458",
  "loyaltyProgramProductSpec": {
    "id": "121",

```

```

    "href" : "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121"
  },
  "loyaltyAccount": {
    "id": "1",
    "href" : "http://server:port/loyaltyManagement/loyaltyProgramMember/121/loyaltyAccount/1"
  },
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1213",
  "validFor": {
    "endDateTime": "2017-05-19T16:42:20Z",
    "startDateTime": "2016-04-19T12:00:00Z"
  }
}

```

DELETE

/loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation is used to delete a loyalty program product.
- This resource represents a managed entity.
- The ID may be a string (or a string containing numbers).

Behavior :

Returns status codes

- 200 OK – The loyalty program product was deleted.
- 404 Not Found – The loyalty program product could not be found.

REQUEST

DELETE /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1213
Accept: application/json

RESPONSE

200
Content-Type: application/json

GET /loyaltyManagement/loyaltyProgramMember/{ID}

This Uniform Contract operation is used to retrieve the representation of a managed entity or a task.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyProgramMember with no {ID}

Description :

- This operation retrieves loyalty program members. The loyalty program member contains accounts and balances in logical containers.
- The resource represents a managed entity or a collection depending on the query pattern.
- The identifier is a string that can consist of numbers and letters.
- Filtering is enabled on all the first level loyalty program member attributes.
- Attribute selection is enabled on all the first level loyalty program member attributes.

Behavior :

Return status codes

- 200 OK - The request was successful (includes situation in which no loyalty program members match the filtering criteria)
- 404 Not Found – The loyalty member could not be found. This return code is only applicable when a managed entity is requested.

REQUEST
GET /loyaltyManagement/loyaltyProgramMember/JDSU778DS Accept: application/json
RESPONSE
200 Content-Type: application/json { "id": "JDSU778DS", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS", "status": "active", "name": "Jane Joe", "validFor": { "startDateTime": "2015-04-19T16:42:23.OZ", "endDateTime": "2016-04-19T16:42:23.OZ" }, "loyaltyAccount": [{ "id": "LoyaltyFund", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS/loyaltyAccount/LoyaltyFund",

```

"loyaltyProgramProduct": {
  "id": "1211",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS/
loyaltyProgramProduct/1211"
},
"loyaltyBalance":{
  "id":"iTunes",
  "balance":1020,
  "validFor": {
    "startDateTime":"2015-04-19T16:42:23.OZ",
    "endDateTime":"2016-04-19T16:42:23.OZ"
  },
  "unit":"magicPoints",
  "loyaltyEarn":{
    "quantity":1000,
    "dateTime":"2016-04-19T16:42:23.OZ",
    "openingBalance":800,
    "closingBalance": 200
  },
  "loyaltyBurn":[]
}
}],
"loyaltyProgramProduct": [{
  "id": "1211",
  "name": "DataUsageBenefit",
  "description": "Data Usage Loyalty Benefits",
  "productStatus": "suspended",
  "productSerialNumber": "S2345666",
  "loyaltyProgramProductSpec": {
    "id": "121",
    "href" : "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121"
  },
  "loyaltyAccount": {
    "id": "LoyaltyFund",
    "href" : "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS/loyaltyAccount/
LoyaltyFund"
  },
  "validFor": {

```



```

    "endDateTime": "2015-05-19T16:42:20Z",
    "startDateTime": "2017-05-19T16:42:20Z"
  },
  "href": " http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS/
loyaltyProgramProduct/1211"
}]
}

```

PUT /loyaltyManagement/loyaltyProgramMember/{ID}

PUT is not supported for loyaltyProgramMember since the loyaltyProgramProduct and loyaltyAccount resources cannot be updated on the endpoint.

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- This operation updates the attributes of a loyalty program member present in the request body.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters.
- Only first level attributes can be updated in its entirety. The loyalty member account may only be updated by addressing the loyalty member account endpoint.

Behavior :

- 200 OK – The loyalty program member was successfully updated.
- 404 Not Found – The loyalty program member could not be found.
- 422 Unprocessable Entity – There was a field rule violation.

Attribute name	Patchable	Rule
id	N	
status	Y	
name	Y	
validFor.startDateTime	Y	

validFor:endDateTime	Y	The end date must be after the start date.
loyaltyAccount	N	Managed on the loyalty account endpoint.
loyaltyEarn	N	Managed on the loyalty earn endpoint.
loyaltyBurn	N	Managed on the loyalty burn endpoint.

REQUEST

PATCH /loyaltyManagement/loyaltyProgramMember/JDSU778DS
Content-type: application/json

```
{
  "status": "suspended",
  "name": "James Joe"
}
```

RESPONSE

201
Content-Type: application/json

```
{
  "id": "JDSU778DS",
  "status": "suspended",
  "name": "James Joe",
  "validFor": {
    "startDateTime": "2015-04-19T16:42:23.0Z ",
    "endDateTime": "2016-04-19T16:42:23.0Z"
  },
  "loyaltyAccounts": [{
    "id": "LoyaltyFund",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMber/104/loyaltyAccount/LoyaltyFund",
    "loyaltyProgramProduct": {
      "id": "1211",
      "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyProgramProduct/1211"
    }
  ]
}
```

```

    },
    "loyaltyBalance": {
      "id": "iTunes",
      "balance": 1020,
      "validFor": {
        "startDateTime": "2015-04-19T16:42:23.0Z",
        "endDateTime": "2016-04-19T16:42:23.0Z"
      },
      "unit": "magicPoints",
      "loyaltyEarn": {
        "quantity": 1000,
        "dateTime": "2016-04-19T16:42:23.0Z",
        "openingBalance": 800,
        "closingBalance": 200
      }
    }
  ]
}],
"loyaltyProgramProduct": [
  {
    "id": "1211",
    "name": "DataUsageBenefit",
    "description": "Data Usage Loyalty Benefits",
    "productStatus": "suspended",
    "productSerialNumber": "S2345666",
    "loyaltyProgramProductSpec": {
      "id": "121",
      "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121"
    },
    "loyaltyAccount": {
      "id": "LoyaltyFund",
      "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS/loyaltyAccount/
LoyaltyFund"
    },
    "validFor": {
      "endDateTime": "2015-05-19T16:42:20Z",
      "startDateTime": "2017-05-19T16:42:20Z"
    },
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS/
loyaltyProgramProduct/1211"
  }
]

```

```
}
}
```

POST /loyaltyManagement/loyaltyProgramMember

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation creates a new loyalty program member.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters.
- There are no mandatory attributes to create a new loyalty program member. If no identifier is provided, an identifier is autogenerated.

Behavior :

- 201 Created – The loyalty program member was successfully created.
- 409 Conflict – A loyalty program member with the same identifier already exists.
- 422 Unprocessable Entity – There was a field rule violation.

Specify the attributes required when an entity is created (and their default values if not):

Attribute name	Mandatory	Default	Rule
id	N	Auto generated if not present in the request.	
status	N	Empty string	
name	N	Empty string	
validFor.startDateTime	N	Today	Valid date not earlier than one bill cycle in the past
validFor.endDateTime	N		After the start date

REQUEST

POST /loyaltyManagement/loyaltyProgramMember
Content-type: application/json

```
{
  "status": "suspended",
  "name": "James Joe",
  "validFor": {
    "startDateTime": "2015-04-19T16:42:23.0Z",
    "endDateTime": "2016-04-19T16:42:23.0Z"
  }
}
```

RESPONSE

201
Content-Type: application/json
Content-Location: /loyaltyManagement/loyaltyProgramMember/123

```
{
  "id": "123",
  "status": "suspended",
  "name": "James Joe",
  "validFor": {
    "startDateTime": "2015-04-19T16:42:23.0Z",
    "endDateTime": "2016-04-19T16:42:23.0Z"
  },
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/123",
  "loyaltyAccount": [],
  "loyaltyProgramProduct": []
}
```

DELETE /loyaltyManagement/loyaltyProgramMember/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation deletes a loyalty program member.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters.

Behavior :

- 200 OK – The loyalty program member was successfully deleted.
- 404 Not Found – The loyalty program member could not be found.

REQUEST
DELETE /loyaltyManagement/loyaltyProgramMember/JDSU778DS
RESPONSE
200

GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount

This Uniform Contract operation is used to retrieve the representation of a managed entity or a task.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount with no {ID}

Description :

- This operation retrieves loyalty program member accounts. The account resource is also returned within each balance entity.
- The resource represents a managed entity or a collection depending on the query pattern.
- The identifier is a string that can consist of numbers and letters.
- Filtering is enabled on all the first level loyalty account attributes.
- Attribute selection is enabled on all the first level loyalty account attributes.

Behavior :

Return status codes

- 200 OK - The request was successful (includes situation in which no loyalty program accounts match the filtering criteria)
- 404 Not Found – The loyalty member or loyalty account could not be found.

REQUEST
GET /loyaltyManagement/loyaltyProgramMember/JDSU778DS/loyaltyAccount Accept: application/json
RESPONSE
200 Content-Type: application/json

```

[[
  "id": "ValueBundle",
  "href": " http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/
loyaltyAccount/ValueBundle",
  "loyaltyProgramProduct": {
    "id": "1211",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/
loyaltyProgramProduct/1211"
  },
  "loyaltyBalance": [{
    "id": "iTunes",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/
loyaltyBalance/iTunes",
    "unit": "NZD",
    "balance": 300,
    "validFor": {
      "startDateTime": "2016/02/19 18:42:23",
      "endDateTime": "2018/12/30 17:42:23"
    }
  }
]]

```

PUT /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}

PUT is not supported for loyaltyAccount as no attribute may be updated.

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}

PATCH is not supported for loyaltyAccount as no attribute may be updated.

POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation creates a new loyalty program member account to contain loyalty balances. A loyalty account corresponds to a existing loyalty program product to which the loyalty member subscribes.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters.
- There are only mandatory attributes for a loyalty account is an existing programProductId.

Behavior :

- 201 Created – The loyalty program account was successfully created.
- 409 Conflict – A loyalty program account with the same identifier already exists.
- 422 Unprocessable Entity – There was a field rule violation.

Specify the attributes required when an entity is created (and their default values if not):

Attribute name	Mandatory	Default	Rule
id	N	Auto generated if not present in the request.	
loyaltyProgramProductId	Y		Must be an existing product belonging to the loyalty program member.

REQUEST

POST /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount
Content-type: application/json

```
{
  "id": "JohnLoyaltyAccount",
  "loyaltyProgramProductId": "1211"
}
```

RESPONSE

201
Content-Type: application/json
Content-Location: /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/
JohnLoyaltyAccount

```
{
  "id": "JohnLoyaltyAccount",
  "href": " http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/"
}
```



```

JohnLoyaltyAccount",
  "loyaltyProgramProduct": {
    "id": "1211",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/
loyaltyProgramProduct/1211"
  },
  "loyaltyBalance": []
}

```

DELETE /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation deletes a loyalty program member account.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters.

Behavior :

- 200 OK – The loyalty program member account was successfully deleted.
- 404 Not Found – The loyalty program member or loyalty program member account could not be found.

REQUEST

```
DELETE /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/JohnLoyaltyAccount
```

RESPONSE

```
200
```

GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance

This Uniform Contract operation is used to retrieve the representation of a managed entity or a task.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance with no {ID}.

Description :

- This operation retrieves loyalty program member account balances. The balance resource contains loyalty earn and loyalty burn collections.
- The resource represents a managed entity or a collection depending on the query pattern.

- The identifier is a string that can consist of numbers and letters.
- Filtering is enabled on all the first level loyalty balance attributes.
- Attribute selection is enabled on all the first level loyalty balance attributes.

Behavior :

Return status codes

- 200 OK - The request was successful (includes situation in which no loyalty program balance match the filtering criteria)
- 404 Not Found – The loyalty member or loyalty balance could not be found.

REQUEST
GET /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance Accept: application/json
RESPONSE
200 Content-Type: application/json <pre> [[{ "id": "iTunes", "href": " http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes", "unit": "NZD", "balance": 300.00, "loyaltyAccount": { "id": "ValueBundle", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/ValueBundle " } }, { "validFor": { "startDateTime": "2016/02/19 18:42:23", "endDateTime": "2018/12/30 17:42:23" }, "loyaltyEarn": [{ "id": "843G-838F-HY23-0238", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/843G-838F-HY23-0238", "quantity": 20.00, </pre>

```

"openingBalance": 280.00,
"closingBalance": 300.00,
"dateTime": "2016-07-29 12:18:51",
"description": "A loyalty event triggered and points earned."
},
{
  "id": "146G-3408-WW40-P238",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/146G-3408-WW40-P238",
  "quantity": 10.00,
  "openingBalance": 250.00,
  "closingBalance": 260.00,
  "dateTime": "2016-07-12 23:34:25",
  "description": "Earn loyalty points on Justin Bieber album purchase."
}],
"loyaltyBurn": [{
  "id": "94JU-03J8-57S4-0893",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/94JU-03J8-57S4-0893",
  "quantity": 20.00,
  "openingBalance": 1000.00,
  "closingBalance": 1020.00,
  "dateTime": "2016-07-08 13:11:27",
  "description": "Beyonce CD purchase loyalty points."
}],
{
  "id": "0347-4378-HT82-6CSJ",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/0347-4378-HT82-6CSJ",
  "quantity": 179.00,
  "openingBalance": 3430.00,
  "closingBalance": 3251.00,
  "dateTime": "2016-07-08 17:15:41",
  "description": "Burn loyalty points on Beyonce Song purchase."
}]
}]

```

PUT /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}

PUT is not supported for loyaltyBalance since the loyaltyAccount resource and the balance cannot be updated on the endpoint.

PATCH /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}

This Uniform Contract operation is used to partially update the representation of a managed entity or a task.

Description :

- This operation updates the attributes of a loyalty program member loyalty balance present in the request body.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters.
- Only first level attributes can be updated in its entirety. loyaltyEarn and loyaltyBurn resources are managed on the respective resource endpoints.

Behavior :

- 200 OK – The loyalty program member loyalty balance was successfully updated.
- 404 Not Found – The loyalty program member or loyalty balance could not be found.
- 422 Unprocessable Entity – There was a field rule violation.

Attribute name	Patchable	Rule
id	N	
unit	Y	
balance	N	
validFor.startDateTime	Y	
validFor.endDateTime	Y	The end date must be after the start date.
loyaltyAccount	N	Managed on the loyalty account endpoint.
loyaltyEarn	N	Managed on the loyalty earn endpoint.

loyaltyBurn	N	Managed on the loyalty burn endpoint.
-------------	---	---------------------------------------

REQUEST

PATCH /loyaltyManagement/loyaltyProgramMember/JDSU778DS
Content-type: application/json

```
{
  "validFor": {
    "startDateTime": "2016/02/19 18:42:23",
    "endDateTime": "2018/12/30 17:42:23"
  }
}
```

RESPONSE

201
Content-Type: application/json

```
{
  "id": "iTunes",
  "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes",
  "unit": "NZD",
  "balance": 300.00,
  "loyaltyAccount": {
    "id": "ValueBundle",
    "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/ValueBundle "
  },
  "validFor": {
    "startDateTime": "2016/02/19 18:42:23",
    "endDateTime": "2018/12/30 17:42:23"
  },
  "loyaltyEarn": [{
    "id": "843G-838F-HY23-0238",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/843G-838F-HY23-0238",
    "quantity": 20.00,
    "openingBalance": 280.00,
    "closingBalance": 300.00,
  }
]
```

```

    "dateTime": "2016-07-29 12:18:51",
    "description": "A loyalty event triggered and points earned."
  },
  {
    "id": "146G-3408-WW40-P238",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/146G-3408-WW40-P238",

    "quantity": 10.00,
    "openingBalance": 250.00,
    "closingBalance": 260.00,
    "dateTime": "2016-07-12 23:34:25",
    "description": "Earn loyalty points on Justin Bieber album purchase."
  }
}],
"loyaltyBurn": [{
  "id": "94JU-03J8-57S4-0893",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/94JU-03J8-57S4-0893",
  "quantity": 20.00,
  "openingBalance": 1000.00,
  "closingBalance": 1020.00,
  "dateTime": "2016-07-08 13:11:27",
  "description": "Beyonce CD purchase loyalty points."
}],
{
  "id": "0347-4378-HT82-6CSJ",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/0347-4378-HT82-6CSJ",
  "quantity": 179.00,
  "openingBalance": 3430.00,
  "closingBalance": 3251.00,
  "dateTime": "2016-07-08 17:15:41",
  "description": "Burn loyalty points on Beyonce Song purchase."
}]
}]
}

```

POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation creates a new loyalty program member account balance.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters.
- There are only mandatory attributes for a loyalty balance is an existing loyalty account.

Behavior :

- 201 Created – The loyalty program balance was successfully created.
- 409 Conflict – A loyalty program balance with the same identifier already exists.
- 422 Unprocessable Entity – There was a field rule violation.

Specify the attributes required when an entity is created (and their default values if not):

Attribute name	Mandatory	Default	Rule
id	N	Auto generated if not present in the request.	
unit	Y		
balance	N	0.00	
validFor.startDateTime	N	Today	
validFor.endDateTime	N		Must be after the start date.
loyaltyAccountId	Y		The loyalty account must exist.

REQUEST

POST /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance
Content-type: application/json

```
{
  "id": "839",
  "loyaltyAccountId": "JohnLoyaltyAccount",
  "unit": "points"
}
```

}
RESPONSE
201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/839 <pre>{ "id": "839", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/839", "balance": 0.00, "validFor": { "startDateTime": "2016-07-08 17:15:41" }, "loyaltyAccount": { "id": "JohnLoyaltyAccount", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyAccount/JohnLoyaltyAccount " } }</pre>

DELETE /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount/{ID}

This Uniform Contract operation is used to delete a managed entity or a task.

Description :

- This operation deletes a loyalty program member balance.

Behavior :

- 200 OK – The loyalty program member balance was successfully deleted.
- 404 Not Found – The loyalty program member or loyalty program balance could not be found.

REQUEST
DELETE /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/839
RESPONSE
200

GET /loyaltyManagement/loyaltyProgramMember/{ID} /loyaltyBalance/{ID}/loyaltyEarn

This Uniform Contract operation is used to retrieve a collection of a managed entity.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}/loyaltyEarn with no {ID}

Description :

- This operation retrieves a loyalty program member loyalty earn transactions on a specific account balance.
- The resource presents a collection of a managed entity, depending on the query pattern.
- The identifier is a string that can consist of numbers and letters.
- Filtering is enabled on all the first level loyalty earn attributes.
- Attribute selection is enabled on all the first level loyalty earn attributes.

Behavior :

Return status codes

- 200 OK - The request was successful (includes situation in which no loyalty program balance match the filtering criteria)
- 404 Not Found – The loyalty member, loyalty balance or loyalty earn transaction could not be found.

REQUEST

GET /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn
Accept: application/json

RESPONSE

```
200
Content-Type: application/json

[[
  {
    "id": "843G-838F-HY23-0238",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/843G-838F-HY23-0238",
    "quantity": 20.00,
    "openingBalance": 280.00,
    "closingBalance": 300.00,
    "dateTime": "2016-07-29 12:18:51",
    "description": "A loyalty event triggered and points earned."
  }
]]
```

```

},
{
  "id": "146G-3408-WW40-P238",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/146G-3408-WW40-P238",
  "quantity": 10.00,
  "openingBalance": 250.00,
  "closingBalance": 260.00,
  "dateTime": "2016-07-12 23:34:25",
  "description": "Earn loyalty points on Justin Bieber album purchase."
}
}

```

POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}/loyaltyEarn

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation creates a new loyalty earn transaction.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters. The identifier serves as the transaction identifier.
- The only mandatory attribute to create a loyalty earn transaction is quantity.

Behavior :

- 201 Created – The loyalty earn transaction was successfully created.
- 409 Conflict – A loyalty transaction with the same identifier already exists.
- 422 Unprocessable Entity – There was a field rule violation.

Specify the attributes required when an entity is created (and their default values if not):

Attribute name	Mandatory	Default	Rule
id	N	Auto generated if not present in the request.	
quantity	Y		The quantity may be of type string or number. If it is a string, it will be converted to a number. If it cannot be converted, a 409 error will

			be returned.
description	N	Empty string	

REQUEST

POST /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn
Content-type: application/json

```
{
  "quantity": 30,
  "description": "Earned loyalty points on handset purchase."
}
```

RESPONSE

201
Content-Type: application/json

```
{
  "id": "738F-039J-2636-LDH8",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyEarn/738F-039J-2636-LDH8",
  "quantity": 30,
  "openingBalance": 280.00,
  "closingBalance": 310.00,
  "dateTime": "2016-07-29 12:18:51",
  "description": "Earned loyalty points on handset purchase."
}
```

GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}/loyaltyBurn

This Uniform Contract operation is used to retrieve a collection of a managed entity.

Note that collections can be retrieved via GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}/loyaltyBurn with no {ID}

Description :

- This operation retrieves a loyalty program member loyalty burn transactions on a specific account balance.
- The resource presents a collection or a managed entity, depending on the query pattern.

- The identifier is a string that can consist of numbers and letters.
- Filtering is enabled on all the first level loyalty burn attributes.
- Attribute selection is enabled on all the first level loyalty burn attributes.

Behavior :

Return status codes

- 200 OK - The request was successful (includes situation in which no loyalty program balance match the filtering criteria)
- 404 Not Found – The loyalty member, loyalty balance or loyalty earn transaction could not be found.

REQUEST
GET /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>[{ "id": "94JU-03J8-57S4-0893", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/94JU-03J8-57S4-0893", "quantity": 20.00, "openingBalance": 1000.00, "closingBalance": 1020.00, "dateTime": "2016-07-08 13:11:27", "description": "Beyonce CD purchase loyalty points." }, { "id": "0347-4378-HT82-6CSJ", "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/0347-4378-HT82-6CSJ", "quantity": 179.00, "openingBalance": 3430.00, "closingBalance": 3251.00, "dateTime": "2016-07-08 17:15:41", "description": "Burn loyalty points on Beyonce Song purchase." }]</pre>

```
}]
```

POST /loyaltyManagement/loyaltyProgramMember/{ID} /loyaltyBalance/{ID}/loyaltyBurn

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation creates a new loyalty burn transaction.
- The resource represents a managed entity.
- The identifier is a string that can consist of numbers and letters. The identifier serves as the transaction identifier.
- The only mandatory attribute to create a loyalty burn transaction is quantity.

Behavior :

- 201 Created – The loyalty burn transaction was successfully created.
- 409 Conflict – A loyalty transaction with the same identifier already exists.
- 422 Unprocessable Entity – There was a field rule violation.

Specify the attributes required when an entity is created (and their default values if not):

Attribute name	Mandatory	Default	Rule
id	N	Auto generated if not present in the request.	
quantity	Y		The quantity may be of type string or number. If it is a string, it will be converted to a number. If it cannot be converted, a 409 error will be returned.
description	N	Empty string	

REQUEST

```
POST /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn
Content-type: application/json
```

```
{
```

```

"quantity": 20,
"description": "Burned loyalty points on album purchase."
}

```

RESPONSE

```

201
Content-Type: application/json

{
  "id": "738F-039J-2636-LDH8",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/iTunes/loyaltyBurn/738F-039J-2636-LDH8",
  "quantity": 20,
  "openingBalance": 310.00,
  "closingBalance": 290.00,
  "dateTime": "2016-08-05 12:18:51",
  "description": "Burned loyalty points on album purchase."
}

```

POST /loyaltyManagement/loyaltyEvent

This Uniform Contract operation is used to create a managed entity or a task.

Description :

- This operation specifies a loyalty event that has occurred.
- The resource only triggers the loyalty rule evaluation for the incoming event's loyalty member, if the event type corresponds to a loyalty rule associated with the loyalty member's products.
- The only mandatory attribute in the loyalty event request is eventType.

Behavior :

- 201 Created – The loyalty event was successfully received.
- 422 Unprocessable Entity – There was a field rule violation.

Specify the attributes required when an entity is created (and their default values if not):

Attribute name	Mandatory	Default	Rule
eventType	Y		

REQUEST

POST /loyaltyManagement/loyaltyEvent
Content-type: application/json

```
{
  "eventId": "00001",
  "eventTime": "2013-04-19T16:42:25-04:00",
  "eventType": "orderCreationNotification",
  "event": {
    "productOrder": {
      "id": "42",
      "href": "http://serverlocation:port/orderManagement/productOrder/42",
      "externalId": "NiceNameForTheConsumer_42",
      ...
    }
  }
}
```

RESPONSE

201
Content-Type: application/json

API NOTIFICATION TEMPLATES

Register LISTENER POST /loyaltyManagement/loyaltyProgramMember/hub

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Supports multiple listeners.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

REQUEST

POST /loyaltyManagement/loyaltyProgramMember/hub
Accept: application/json

```
{"callback": "http://in.listener.com"}
```

RESPONSE

201
Content-Type: application/json
Location: /loyaltyManagement/loyaltyProgramMember/hub/42

```
{"id":"42","callback":"http://in.listener.com","query":null}
```

Unregister LISTENER DELETE /loyaltyManagement/loyaltyProgramMember/hub/{id}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

REQUEST

DELETE /loyaltyManagement/loyaltyProgramMember/hub/42
Accept: application/json

RESPONSE
204

Publish loyaltyProgramMember POST /listener

Description :

The supported loyalty program member event types are:

- LoyaltyProgramMemberCreationNotification
- LoyaltyProgramMemberUpdateNotification
- LoyaltyProgramMemberDeleteNotification

Each of these events will publish an event to the registered listeners, will request structures are set out in the event model section.

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

REQUEST
POST /client/listener Accept: application/json <pre>{ "eventId": "111", "eventType": "LoyaltyProgramMemberCreationNotification", "event": { EVENT BODY as described in loyaltyProgramMember event model section. } }</pre>
RESPONSE
201 Content-Type: application/json

Register LISTENER POST /loyaltyManagement/loyaltyProgramMemberProduct/hub

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Supports multiple listeners.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

REQUEST

POST /loyaltyManagement/loyaltyProgramMemberProduct/hub
Accept: application/json

```
{"callback": "http://in.listener.com"}
```

RESPONSE

201
Content-Type: application/json
Location: /loyaltyManagement/loyaltyProgramMemberProduct/hub/42

```
{"id":"42","callback":"http://in.listener.com","query":null}
```

Unregister LISTENER DELETE /loyaltyManagement/loyaltyProgramMemberProduct/hub/{id}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

REQUEST

DELETE /loyaltyManagement/loyaltyProgramMemberProduct/hub/42
Accept: application/json

RESPONSE

204

Publish loyaltyProgramMemberProduct POST /listener

Description :

The supported loyalty program member product event types are:

- LoyaltyProgramMemberProductCreationNotification
- LoyaltyProgramMemberProductUpdateNotification
- LoyaltyProgramMemberProductDeleteNotification

Each of these events will publish an event to the registered listeners, will request structures are set out in the event model section.

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

REQUEST

POST /client/listener
Accept: application/json

```
{  
  "eventId": "111",  
  "eventType": "LoyaltyProgramMemberProductCreationNotification",  
  "event": {  
    EVENT BODY as described in loyaltyProgramProductMember event model section.  
  }  
}
```

RESPONSE

201
Content-Type: application/json

Register LISTENER POST /loyaltyManagement/loyaltyEarn/hub

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Supports multiple listeners.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

REQUEST
POST /loyaltyManagement/loyaltyEarn/hub Accept: application/json {"callback": "http://in.listener.com"}
RESPONSE
201 Content-Type: application/json Location: /loyaltyManagement/loyaltyEarn/hub/42 {"id": "42", "callback": "http://in.listener.com", "query": null}

Unregister LISTENER DELETE /loyaltyManagement/loyaltyEarn/hub/{id}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

REQUEST
DELETE /loyaltyManagement/loyaltyEarn/hub/42 Accept: application/json
RESPONSE
204

Publish loyaltyEarn POST /listener

Description :

The only supported loyalty earn event type is loyaltyEarnNotification.

Each of these events will publish an event to the registered listeners, will request structures are set out in the event model section.

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

REQUEST

POST /client/listener
Accept: application/json

```
{
  "eventId": "111",
  "eventType": "LoyaltyEarnNotification",
  "event": {
    EVENT BODY as described in loyaltyEarn event model section.
  }
}
```

RESPONSE

201
Content-Type: application/json

Register LISTENER POST /loyaltyManagement/loyaltyBurn/hub

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Supports multiple listeners.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

REQUEST

POST /loyaltyManagement/loyaltyBurn/hub
Accept: application/json

```
{"callback": "http://in.listener.com"}
```

RESPONSE

```
201
Content-Type: application/json
Location: /loyaltyManagement/loyaltyBurn/hub/42
{"id":"42","callback":"http://in.listener.com","query":null}
```

Unregister LISTENER DELETE /loyaltyManagement/loyaltyBurn/hub/{id}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

REQUEST

```
DELETE /loyaltyManagement/loyaltyBurn/hub/42
Accept: application/json
```

RESPONSE

```
204
```

Publish loyaltyBurn POST /listener

Description :

The only supported loyalty earn event type is loyaltyBurnNotification.

Each of these events will publish an event to the registered listeners, will request structures are set out in the event model section.

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

REQUEST

```
POST /client/listener
```

Accept: application/json

```
{
  "eventId": "111",
  "eventType": "LoyaltyBurnNotification",
  "event": {
    EVENT BODY as described in loyaltyBurn event model section.
  }
}
```

RESPONSE

201
Content-Type: application/json

Register LISTENER POST /loyaltyManagement/loyaltyEvent/hub

Description :

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Supports multiple listeners.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

REQUEST

POST /loyaltyManagement/loyaltyEvent/hub
Accept: application/json

```
{"callback": "http://in.listener.com"}
```

RESPONSE

201
Content-Type: application/json
Location: /loyaltyManagement/loyaltyEvent/hub/42

```
{"id": "42", "callback": "http://in.listener.com", "query": null}
```

Unregister LISTENER DELETE /loyaltyManagement/loyaltyEvent/hub/{id}

Description :

Clears the communication endpoint address that was set by creating the Hub.

Behavior :

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

REQUEST

```
DELETE /loyaltyManagement/loyaltyEvent/hub/42
Accept: application/json
```

RESPONSE

```
204
```

Publish loyaltyEvent POST /listener

Description :

The only supported loyalty earn event type is loyaltyEventNotification.

Each of these events will publish an event to the registered listeners, will request structures are set out in the event model section.

Behavior :

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

REQUEST

```
POST /client/listener
Accept: application/json
```

```
{
  "eventId": "111",
  "eventType": "LoyaltyEventNotification",
  "event": {
    EVENT BODY as described in loyaltyEvent event model section.
  }
}
```


RESPONSE

201
Content-Type: application/json

OPEN ISSUES

There are several open issues that needs to be considered:

- In the GB922_Loyalty_R14.0.pdf document, a loyaltyProgramMember is not directly linked to a loyaltyProgramProduct. Is it acceptable that our API specification links the product instance directly with the loyalty program member?
- In the GB922_Loyalty_R14.0.pdf document, there is a statement, "LoyaltyPrograms might be incompatible with each other and in this case the customer has to choose only one LoyaltyProgram". How will this look in the design model? The document does not elaborate further on this point.
- The SID specifies the loyaltyAction as "A LoyaltyAction may correspond either to a CustomerOrder (Ex: 100 SMS free), or to an Interaction (Ex: a SMS notifying the 100 SMS free) or to a LoyaltyEarn (Ex: 100 points on the LoyaltyAccount)", but the GB922_Loyalty_R14.0.pdf document diagram specifies that a LoyaltyAction may have a CustomerOrder and a LoyaltyEarn and Interaction. Which is correct?
- If there are multiple loyaltyActions and there is a failure during the execution of a loyalty action, there is no rollback procedure. We propose that if there are multiple loyalty actions necessary, that the API consumer specifies the loyaltyAction as a business interaction and handle the rest of the actions necessary himself.
- Should an error be returned when deleting a product specification and there are products that have been created according to this specification?
- In the GB922_Loyalty_R14.0.pdf document, figure Pr. 38 indicates that a loyaltyAction describes multiple loyaltyExecutionPoints. We propose that this be changed to a loyaltyAction only describing a single loyaltyExecutionPoint, since in practice it is unlikely that you will have a single loyaltyAction resource will adequately be able to describe a LoyaltyEarn, CustomerOrder and BusinessInteraction resource.
- Considering the "unit" attribute on the LoyaltyBalance resource, it may be necessary to specify a conversion ratio of loyalty units if it is not a monetary unit.

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	04/15/2013	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of Draft Version of the Document.
Release 1.1			Updated for use in the Paris Spec Jam – and rebranded,.