



## *TM Forum Specification*

# Loyalty Management API REST Specification

**TMF658**  
**Release 17.0.1**  
**November 2017**

<b>Latest Update: TM Forum Release 17</b>	<b>TM Forum Approved</b>
<b>Version 2.0.1</b>	<b>IPR Mode: RAND</b>

## NOTICE

Copyright © TM Forum 2017. 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.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website, but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100  
Parsippany, NJ 07054 USA

Tel No. +1 973 944 5100

Fax No. +1 973 944 5110

TM Forum Web Page: [www.tmforum.org](http://www.tmforum.org)

## TABLE OF CONTENTS

NOTICE.....	2
Table of Contents.....	4
List of figures.....	10
Introduction .....	11
Program Specification Resources .....	11
Program Member Resources.....	12
SPECIFIC Considerations .....	13
eTOM .....	14
SAMPLE USE CASES.....	15
Use Case 1: Administration of Loyalty Program Specification Master Reference Data.....	15
Description.....	15
Main Actors.....	15
Use Case Steps.....	15
Example of API Usage in the Context of the Use Case .....	15
Success Outcome.....	16
Use Case 2: On-boarding of LoyaltyProgramMembers to specific LoyaltyAccount.....	16
Description.....	16
Main Actors.....	16
Use Case Steps.....	16
Example of API Usage in the Context of the Use Case .....	17
Success Outcome.....	17
Use Case 3: Processing of events that may result in a LoyaltyAction being executed.....	17
Description.....	17
Main Actors.....	17
Use Case Steps.....	17
Example of API Usage in the Context of the Use Case .....	18
Success Outcome.....	18
Use Case 4: Burn points as part of a LoyaltyProgramMember CustomerPayment .....	18

Description .....	18
Main Actors .....	19
Use Case Steps .....	19
Example of API Usage in the Context of the Use Case .....	19
Success Outcome.....	19
Use Case 5: Specific Example - CustomerOrder resulting in redeemable loyalty points earnings.....	19
Description .....	19
Main Actors .....	20
Use Case Steps .....	20
Example of API Usage in the Context of the Use Case .....	20
Success Outcome.....	20
RESOURCE MODEL.....	21
MANAGED ENTITY AND TASK RESOURCE MODELS .....	21
LoyaltyProgramProductSpec Resource .....	21
LoyaltyRule Resource.....	22
LoyaltyCondition Resource.....	24
LoyaltyAction Resource .....	24
LoyaltyEventType Resource.....	26
LoyaltyEvent Resource.....	26
LoyaltyProgramProduct Resource.....	27
LoyaltyProgramMember Resource .....	28
LoyaltyAccount Resource .....	29
LoyaltyBalance Resource .....	29
LoyaltyEarn Resource .....	30
LoyaltyBurn Resource .....	31
LoyaltyEvent Resource.....	32
LoyaltyExecutionPoint Resource.....	32
Resource Relationships.....	35
API OPERATIONS.....	40
OPERATIONS ON PROGRAM PRODUCT SPEC.....	40

List / Retrieve Program Product Spec ..... 40

Patch Program Product Spec ..... 41

Create Program Product Spec ..... 42

Delete Program Product Spec ..... 45

OPERATIONS ON RULES ..... 45

    List / Retrieve Rules ..... 45

    Update Rule ..... 47

    Patch Rule ..... 48

    Create Rule ..... 50

    Delete Rule ..... 52

OPERATIONS ON Rule Conditions ..... 53

    List / Retrieve Rule Conditions ..... 53

    Create Rule Condition ..... 54

    Delete Rule Condition ..... 55

OPERATIONS ON Rule Event Types ..... 56

    List / Retrieve Rule Conditions ..... 56

    Create Rule Condition ..... 57

    Delete Rule Condition ..... 58

OPERATIONS ON Rule Actions ..... 58

    List / Retrieve Rule Actions ..... 58

    Create Rule Action ..... 59

    Delete Rule Action ..... 61

OPERATIONS ON CONDITIONS ..... 61

    List / Retrieve Conditions ..... 61

    Update Conditions ..... 62

    Patch Condition ..... 63

    Create Condition ..... 65

    Delete Condition ..... 66

OPERATIONS ON ACTIONS ..... 67

    List / Retrieve Actions ..... 67

Update Action ..... 68

Patch Action..... 70

Create Action ..... 71

Delete Action ..... 74

OPERATIONS ON EVENT TYPES ..... 75

    List / Retrieve Event Types..... 75

    Update Event Type ..... 75

    Patch Event Type ..... 76

    Create Event Type..... 77

    Delete Event Type ..... 79

OPERATIONS ON LOYALTY PROGRAM PRODUCT ..... 79

    List / Retrieve Loyalty Program Product..... 79

    Patch Loyalty Program Product..... 81

    Create Loyalty Program Product ..... 83

    Delete Loyalty Program Product..... 89

OPERATIONS ON LOYALTY EXECUTION POINT ..... 90

    List / Retrieve Loyalty Execution Points ..... 90

OPERATIONS ON LOYALTY PROGRAM MEMBER ..... 92

    List / Retrieve Loyalty Program Members ..... 92

    Update Loyalty Program Members..... 93

    Patch Loyalty Program Members ..... 94

    Create Loyalty Program Members..... 96

    Delete Loyalty Program Members ..... 97

OPERATIONS ON LOYALTY ACCOUNT ..... 98

    List / Retrieve Loyalty Balances ..... 98

OPERATIONS ON LOYALTY BALANCE ..... 99

    List / Retrieve Loyalty Balances ..... 99

    Patch Loyalty Balances ..... 100

    Post Loyalty Balances ..... 102

    Delete Loyalty Balances ..... 104

OPERATIONS ON LOYALTY EVENT ..... 104

    List / Retrieve Loyalty Earn ..... 104

    Create Loyalty Earn ..... 106

OPERATIONS ON LOYALTY BURN ..... 107

    List / Retrieve Loyalty Burn ..... 107

    Create Loyalty Burn ..... 109

OPERATIONS ON LOYALTY EVENT ..... 110

    Create Loyalty Event ..... 110

API NOTIFICATION TEMPLATES ..... 112

    NOTIFICATIONS ON LOYALTY PROGRAM MEMBER ..... 112

        Register Loyalty Program Member Listener ..... 112

        Unregister Loyalty Program Member Listener ..... 112

        Publish Loyalty Program Member Listener ..... 113

    NOTIFICATIONS ON LOYALTY MEMBER PRODUCT ..... 114

        Register Loyalty Member Product Listener ..... 114

        Unregister Loyalty Member Product Listener ..... 114

        Publish Loyalty Member Product Listener ..... 115

    NOTIFICATIONS ON LOYALTY EARN ..... 116

        Register Loyalty Earn Listener ..... 116

        Unregister Loyalty Earn Listener ..... 116

        Publish Loyalty Earn Listener ..... 117

    NOTIFICATIONS ON LOYALTY BURN ..... 118

        Register Loyalty Burn Listener ..... 118

        Unregister Loyalty Burn Listener ..... 118

        Publish Loyalty Burn Listener ..... 119

    NOTIFICATIONS ON LOYALTY EVENT ..... 120

        Register Loyalty Event Listener ..... 120

        Unregister Loyalty Event Listener ..... 120

        Publish Loyalty Event Listener ..... 121

Event Models ..... 123



Loyalty Program Member Notification Model.....	123
loyaltyProgramMemberCreateNotification.....	123
loyaltyProgramMemberUpdateNotification.....	123
loyaltyProgramMemberDeleteNotification.....	124
Loyalty Program Member Product Model.....	125
loyaltyProgramMemberProductCreateNotification.....	125
loyaltyProgramMemberUpdateNotification.....	126
loyaltyProgramMemberDeleteNotification.....	126
Loyalty Earn Model.....	127
loyaltyEarnNotification.....	127
Loyalty Burn Model.....	128
loyaltyBurnNotification.....	128
Loyalty Event Model.....	129
loyaltyEventNotification.....	129
OPEN ISSUES.....	131
Release History.....	132
Company Contact Details.....	133

## LIST OF FIGURES

Figure 1 – Loyalty Program Managed Resource Context	11
Figure 2 – LoyaltyProgramMember resource and contained resources	12
Figure 3 – LoyaltyProgramProductSpec resource model	22
Figure 4 – LoyaltyManagement resource model	35
Figure 5 – LoyaltyAction resource model	36
Figure 6 – LoyaltyProgramProduct resource model	37
Figure 7 – LoyaltyAccount resource model	38
Figure 8 – LoyaltyBalance resource model	39

## INTRODUCTION

The TM Forum 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 TM Forum 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.

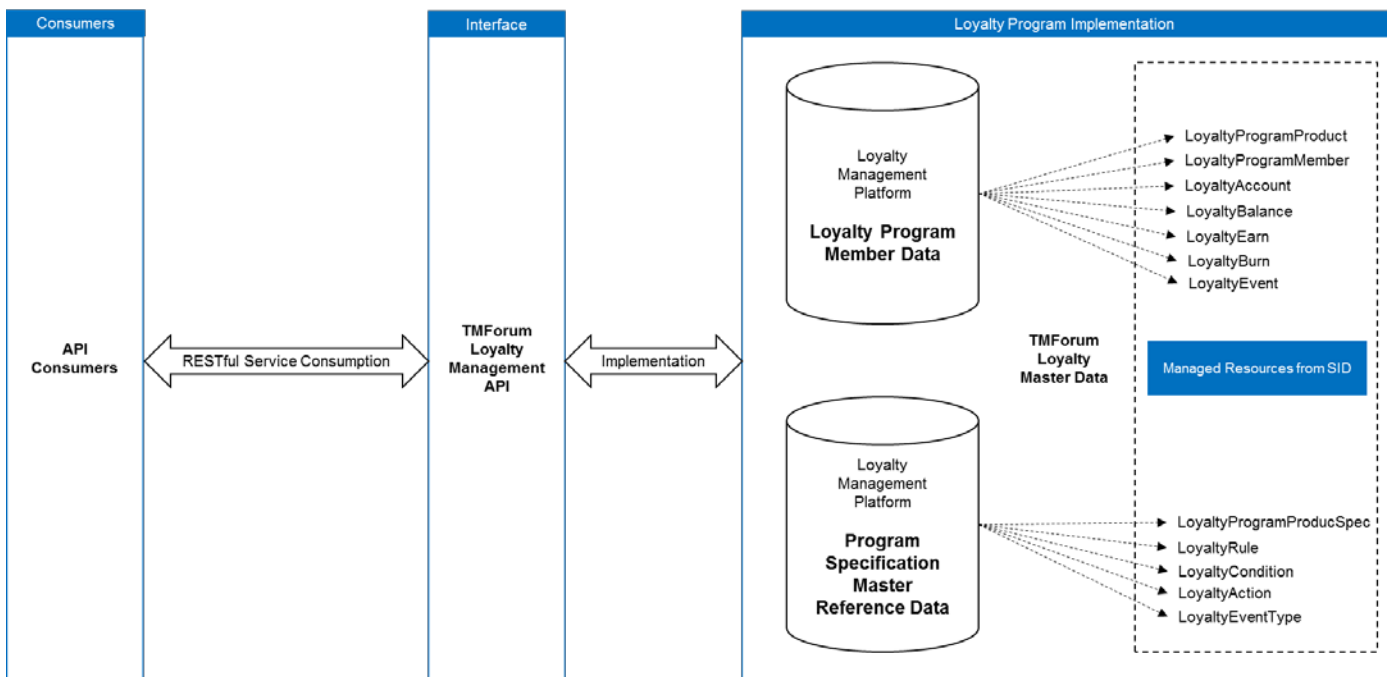


Figure 1 – Loyalty Program Managed Resource Context

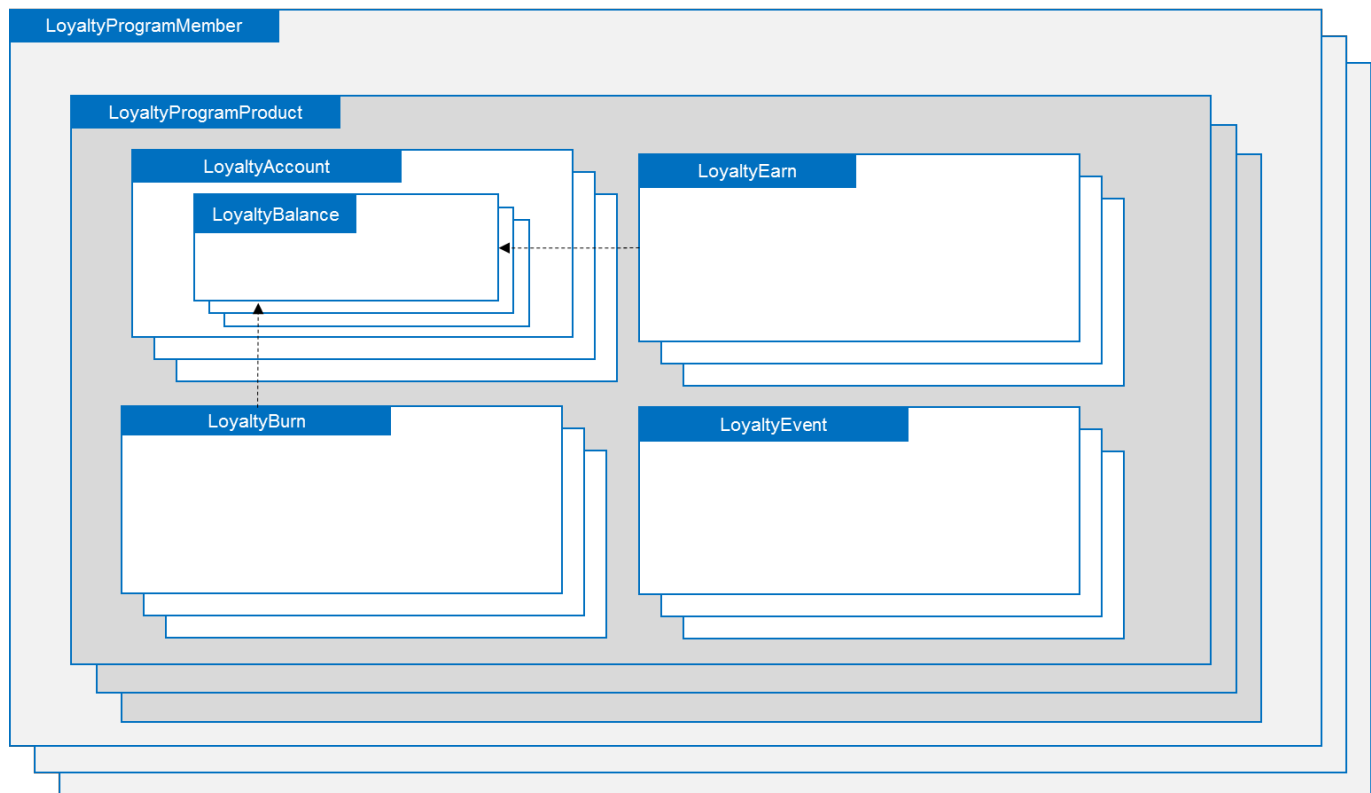
## Program Specification Resources

This grouping of resources defines 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 behavior, 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 2 –LoyaltyProgramMember resource and contained resources**

As depicted in Figure , 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;
- LoyaltyEarn 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 resource<sup>1</sup> 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 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.

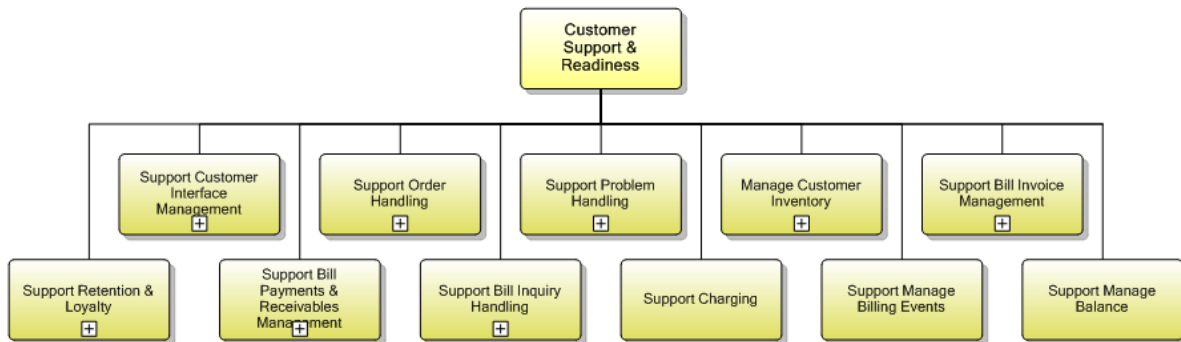
---

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

eTOM

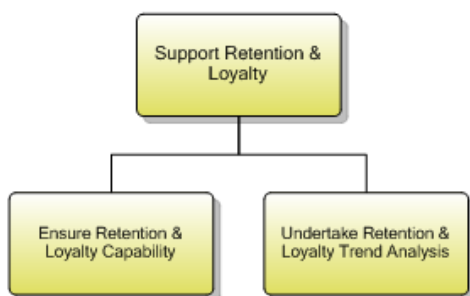
Loyalty and Customer Retention forms part of the Customer Support & Readiness part of eTOM.

**1.3.1 Customer Support & Readiness**



This was created from the Frameworx 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 Frameworx 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 LoyaltyAccount
- 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 customers between the age of 18 and 21 qualify for LoyaltyRule within 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 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; (*POST /loyaltyManagement/loyaltyProgramProductSpec*)
- ii. The User Interface consumes the LoyaltyRule resource to create one or more rules related to the newly created LoyaltyProgramProductSpec; (*POST /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule*)
- 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; (*POST /loyaltyManagement/loyaltyCondition*, *POST /loyaltyManagement/loyaltyRule/{ID}/loyaltyCondition*)
- iv. The User Interface consumes the and produces LoyaltyAction resource to create one or more actions that describe the fulfilment of the loyalty execution. After creation, the action must be linked to the LoyaltyRule; (*POST /loyaltyManagement/loyaltyAction*, *POST /loyaltyManagement/loyaltyRule/{ID}/loyaltyAction*)
- 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; (*POST /loyaltyManagement/loyaltyEventType*, *POST /loyaltyManagement/loyaltyRule/{ID}/loyaltyEventType*)

---

## 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 LoyaltyAccount

---

### Description

The main purpose of this use case is to successfully onboard new Loyalty Program Members to a LoyaltyAccount (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. This corresponds to a CustomerOrder of the LoyaltyProgramProductSpec.



---

## 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. (*GET /loyaltyManagement/loyaltyProgramMember/{ID}*)
- ii. In response to the customer electing to sign up for the loyalty program, the API LoyaltyProgramMember CREATE operation is used to deliver the loyalty program ordered by the customer. The same LoyaltyProgramMember may have rights on many LoyaltyProgram through one or many LoyaltyAccount. 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; (*POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct*)
  - b. The Loyalty Platform can also create a Loyalty Account as part of the program enrolment and perhaps corresponding opening balances. (*POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount*)

---

## 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. A LoyaltyExecutionPoint record is created as event record of the LoyaltyAction executed.

---

## 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 (*POST /loyaltyManagement/loyaltyEvent*);
- ii. The event handling implementation or Loyalty Platform obtains all EventType resources defined as well as their links to LoyaltyRules; (*GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule?loyaltyEventType.eventType=CustomerOrder*)
- 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.
- iv. LoyaltyExecutionPoints may be retrieved using the GET operation.

---

## 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 party LoyaltyProgramMember by performing a GET operation on the LoyaltyProgramMember resource. This operation returns whether or not the customer has rights on a LoyaltyAccount and also returns present account and balance information; (*GET /loyaltyManagement/loyaltyProgramMember/{ID}*, *GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyAccount*)
- 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. (*POST /loyaltyManagement/loyaltyAccount/{ID}/loyaltyBalance/{ID}/loyaltyBurn*)

---

## 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. (*POST /loyaltyManagement/loyaltyEvent*);
- 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. (*GET /loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule?loyaltyEventType.eventType=CustomerOrder, GET /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}* )
- If the 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. (*POST /loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyBalance/{ID}/loyaltyEarn*)

---

## 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 must be checked to identify the actions to apply.

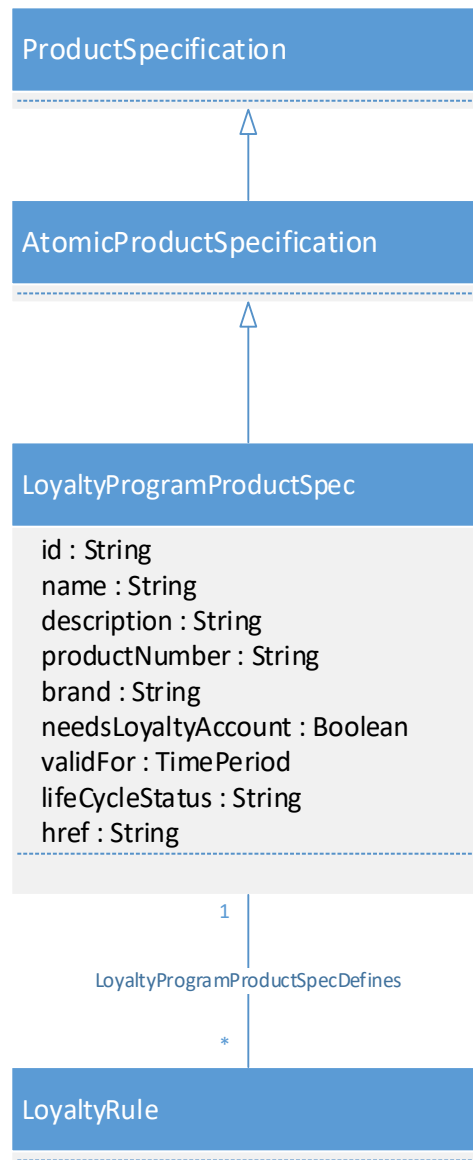
Resource IDs for loyalty program product specifications are strings that may consist of numbers and letters.

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"
  }
}
```

#### Field Descriptions:

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 3– LoyaltyProgramProductSpec resource model**


## 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",
      "href": "/loyaltyManagement/loyaltyEventType/3"
    }],
    "loyaltyCondition": [{
      "id": "1",
      "href": "/loyaltyManagement/loyaltyCondition/1"
    }],
    "loyaltyAction": [{
      "id": "111",
      "href": "/loyaltyManagement/loyaltyAction/111"
    }]
  }

```

### Field Descriptions:

Field	Description
<b>id</b>	Unique identifier for the loyalty rule.
<b>isCNF</b>	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).
<b>hasSubRules</b>	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.
<b>isMandatoryEvaluation</b>	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.
<b>usage</b>	A free-form string attribute that recommends how this policy object should be used.

<b>keywords</b>	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.
<b>policyName</b>	A generic naming attribute that can be used to identify different policy entities.
<b>commonName</b>	A user-friendly identifier of the loyalty rule.
<b>description</b>	A free-form description of the rule.
<b>href</b>	A reference to the loyalty rule.

## LoyaltyCondition Resource

A LoyaltyCondition specifies the condition that is evaluated to decide if the LoyaltyAction(s) from the LoyaltyRule 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"
}
```

### Field Descriptions:

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

## LoyaltyAction Resource

A LoyaltyAction is an action that is 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).



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

The actionAttributes, headers and body sections are attributes that are used when the action point is executed. The actionAttributes are attributes that may be used as replacement attributes in the action endpoint (URL) or in the action body, which is the JSON body that will be send to the endpoint. The headers specified in the headers container is sent as HTTP headers when the HTTP call is made.

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

```
{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "headers": {
    "Content-Type": "text/json"
  },
  "body": {
    "orderType": "dataBundle",
    "orderCatagory": "bundles"
  },
  "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"
}
```

### Field Descriptions:

Field	Description
<b>id</b>	Unique identifier for the loyalty action.
<b>type</b>	The type of loyalty action (LoyaltyEarn, CustomerOrder or BusinessInteraction).
<b>actionAttributes</b>	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.
<b>body</b>	The body if the request that is made when this action is executed. The body is assumed to be of type text/json. Any tokens denoted with {} will be replaced by attributes present in:

	<ul style="list-style-type: none"> <li>• Member profile</li> <li>• actionAttributes</li> <li>• Event attributes in the loyalty event request</li> </ul>
<b>headers</b>	The HTTP headers that is added to the action call.
<b>href</b>	A reference to the loyalty action.
<b>version</b>	A string that identifies the version of the loyalty execution point.
<b>commonName</b>	A user-friendly identifier of the loyalty execution point.
<b>description</b>	A free-form description of the loyalty execution point.
<b>action</b>	The HTTP operation to be used when calling the endpoint (POST, PUT, GET or DELETE )
<b>endpoint</b>	The endpoint to call to trigger a BusinessInteraction, CustomerOrder or LoyaltyEarn.

## 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.

Note that LoyaltyEventType is named LoyaltyEvent in the SID. There is a need to describe the loyalty event types that should be evaluated for possible loyalty rule evaluations. For this reason, the LoyaltyEvent model in the SID is renamed to LoyaltyEventType and LoyaltyEvent denotes the actual loyalty event trigger.

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

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

### Field Descriptions:

## LoyaltyEventType Resource

Field	Description
<b>id</b>	Unique identifier for the loyalty event type.
<b>eventType</b>	The type of loyalty event expected to trigger the loyalty rule evaluation, e.g. a CustomerOrder or an Invoice. The eventType should be unique to prevent existing resources not being reused. An eventType that already exists will result in a 422 error.

## 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. An existing account ID or details for a new LoyaltyAccount and its LoyaltyBalance is specified.

Resource IDs for loyalty program products are strings that may consist of numbers and letters. The product ID only needs to be unique to the member it belongs to.

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

```
{
  "id": "1211",
  "name": "DataUsageBenefit",
  "description": "Data Usage Loyalty Benefits",
  "productStatus": "suspended",
  "href": "/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211",
  "validFor": {
    "endDateTime": "2017-12-19T16:42:20Z",
    "startDateTime": "2017-05-19T16:42:20Z"
  },
  "characteristics": [
    [
      {
        "name": "DataUsageThreshold",
        "value": "10"
      },
      {
        "name": "DataUsageThresholdUnitOfMeasure",
        "value": "GB"
      }
    ]
  ],
  "loyaltyProgramProductSpec": {
    "id": "121",
    "href": "/loyaltyManagement/loyaltyProgramProductSpec/121"
  },
  "loyaltyAccount": {
    "id": "1",
    "href": "/loyaltyManagement/loyaltyProgramMember/2222/loyaltyAccount/1"
  }
}
```

```
}
}
```

### Field Descriptions:

Field	Description
<b>id</b>	Unique identifier for the loyalty program product.
<b>name</b>	A word, term, or phrase by which the product is known and distinguished from other products.
<b>description</b>	An explanation of what the product is.
<b>productStatus</b>	The condition of the product, such as planned, designed, activated, disconnected.
<b>characteristics</b>	Name value pairs of product characteristics.
<b>validFor</b>	The period during which the product is applicable.
<b>loyaltyProgramProductSpec</b>	The product specification that describes this product.
<b>loyaltyAccount</b>	The loyalty account to which results can be logged.
<b>href</b>	A reference to the loyalty program product.

## 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"
  }
}
```

### Field Descriptions:

Field	Description
<b>id</b>	Unique identifier for the loyalty program member.

<b>name</b>	The loyalty member's name.
<b>status</b>	A free-form field in which the member's status may be captured, e.g. "active", "suspended", .etc.
<b>validFor</b>	The period during which the loyalty member is valid for.
<b>href</b>	A reference to the loyalty program member.

## LoyaltyAccount Resource

A LoyaltyAccount corresponds to a set of balances to carry loyalty movement according to different valid period and unit. The LoyaltyAccount is a container for one or more LoyaltyBalances.

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. Account ID's are assumed to be unique over all accounts.

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"
  }
}
```

### Field Descriptions:

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

## 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/loyaltyAccount/JohnLoyalty/loyaltyBalance/iTunes",
  "quantity": {
    "unit": "NZD",
    "balance": 300
  },
  "validFor": {
    "startDateTime": "2016/02/19 18:42:23",
    "endDateTime": "2018/12/30 17:42:23"
  }
}
```

### Field Descriptions:

Field	Description
<b>id</b>	Unique identifier for the loyalty balance.
<b>quantity.unit</b>	Unit of the quantity credited and debited from the balance.
<b>quantity.balance</b>	The current balance of the loyalty balance account.
<b>validFor</b>	The validity period in which loyalty can be burned using the account.
<b>href</b>	A reference to the loyalty member's loyalty balance.

### 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": "2017-05-05T07:00:01Z",
"description": "A loyalty event triggered and points earned."
}

```

### Field Descriptions:

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

## 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": "2017-05-08T07:00:01Z ",
  "description": "Burned loyalty points on album purchase."
}

```

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

<b>openingBalance</b>	The closing quantity on the account balance, after the loyalty units were burned.
<b>dateTime</b>	The date time on which the loyalty burn transaction occurred.
<b>description</b>	A free-form description describing the loyalty burn event.

## 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. The memberId is also mandatory, so to link the event with a loyalty program member.

```

{
  "eventType": "orderCreationNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "4343",
  "memberId": "43403423",
  "event": {
    "productOrder": {
      ...
    }
  }
}

```

Field	Description
<b>eventId</b>	Unique identifier for the loyalty event.
<b>eventType</b>	The incoming loyalty event type.
<b>eventTime</b>	The time that the event was received.
<b>memberId</b>	The loyalty program member ID associated with the event.
<b>event</b>	The actual event container.
<b>Field</b>	Description
<b>eventId</b>	Unique identifier for the loyalty event.

## LoyaltyExecutionPoint Resource

A LoyaltyAction describes a LoyaltyExecutionPoint. A LoyaltyExecutionPoint is a type of PolicyExecutionPoint. A LoyaltyExecutionPoint tracks the application of a LoyaltyAction and serves as a record of an applied LoyaltyAction.



Below is a representation of the LoyaltyExecutionPoint resource in JSON format. The LoyaltyExecutionPoint contains all information used the LoyaltyAction execution, so that an accurate history of applied events is available without being dependent on the LoyaltyAction (which may have been updated or altered).

```
{
  "id": "1243",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "headers": {
    "Content-Type": "text/json"
  },
  "body": {
    "orderType": "dataBundle",
    "orderCategory": "bundles"
  },
  "commonName": "Earn50",
  "description": "Earn loyalty points",
  "action": "POST",
  "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/loyaltyAccount/{accountId}/loyaltyBalance/{balanceId}/loyaltyEarn",
  "href": "/loyaltyManagement/loyaltyExecutionPoint/1243",
  "version": "2.0",
  "dateTime": "2017-05-05T07:00:01Z",
}
```

Field	Description
<b>id</b>	Unique identifier for the loyalty execution point. This maps to the objectId in the SID and is simplified to "id".
<b>type</b>	The type of loyalty action (LoyaltyEarn, CustomerOrder or BusinessInteraction).
<b>actionAttributes</b>	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.
<b>body</b>	The HTTP body sent with the request.
<b>headers</b>	The HTTP headers sent with the request.
<b>href</b>	A reference to the loyalty action.
<b>version</b>	A string that identifies the version of the loyalty execution point.
<b>commonName</b>	A user-friendly identifier of the loyalty execution point.
<b>description</b>	A free-form description of the loyalty execution point.

---

<b>action</b>	The HTTP operation to be used when calling the endpoint (POST, PUT, GET or DELETE )
<b>endpoint</b>	The endpoint to call to trigger a BusinessInteraction, CustomerOrder or LoyaltyEarn.
<b>dateTime</b>	The timestamp that the action was applied.

---

## Resource Relationships

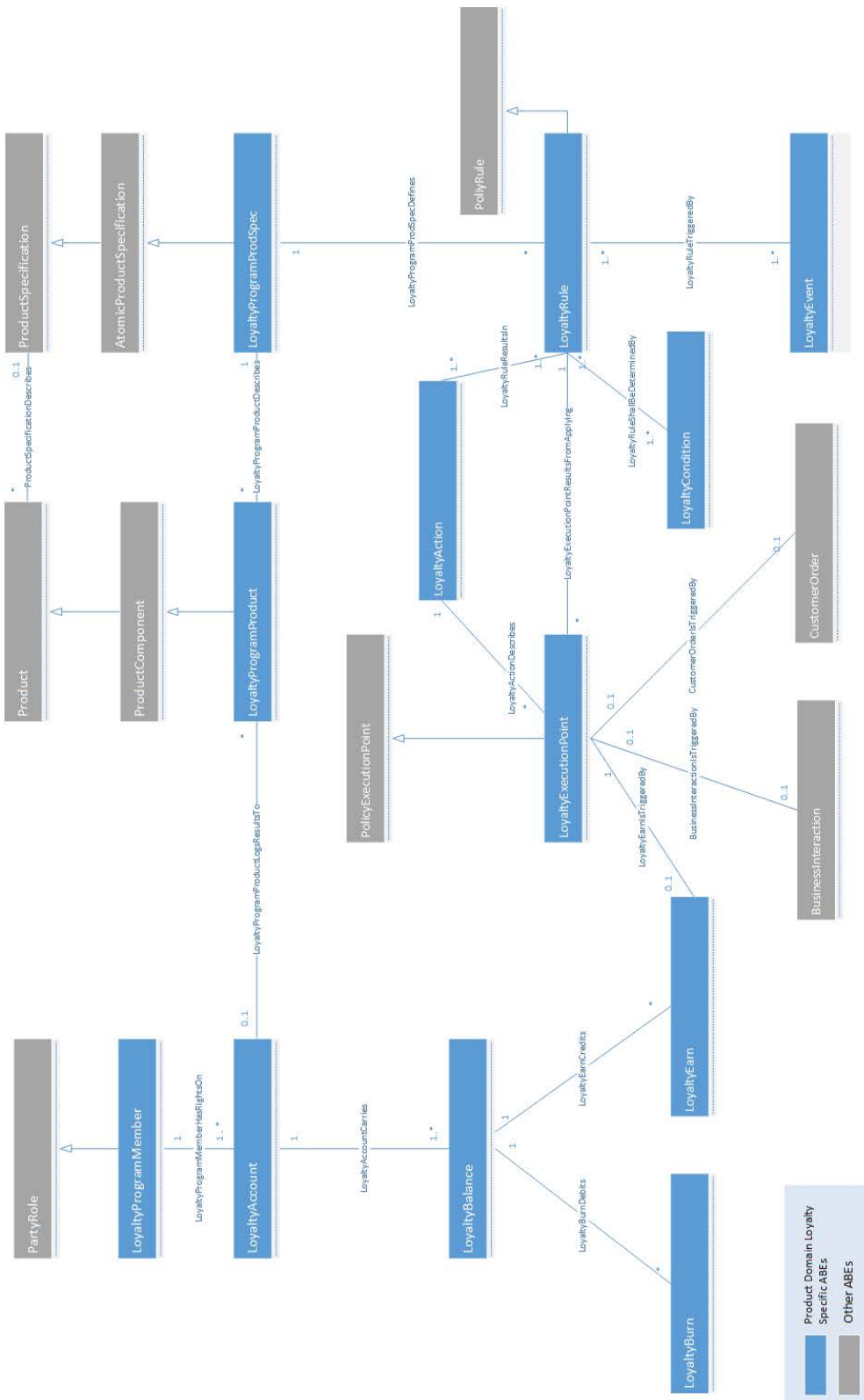


Figure 4– LoyaltyManagement resource model

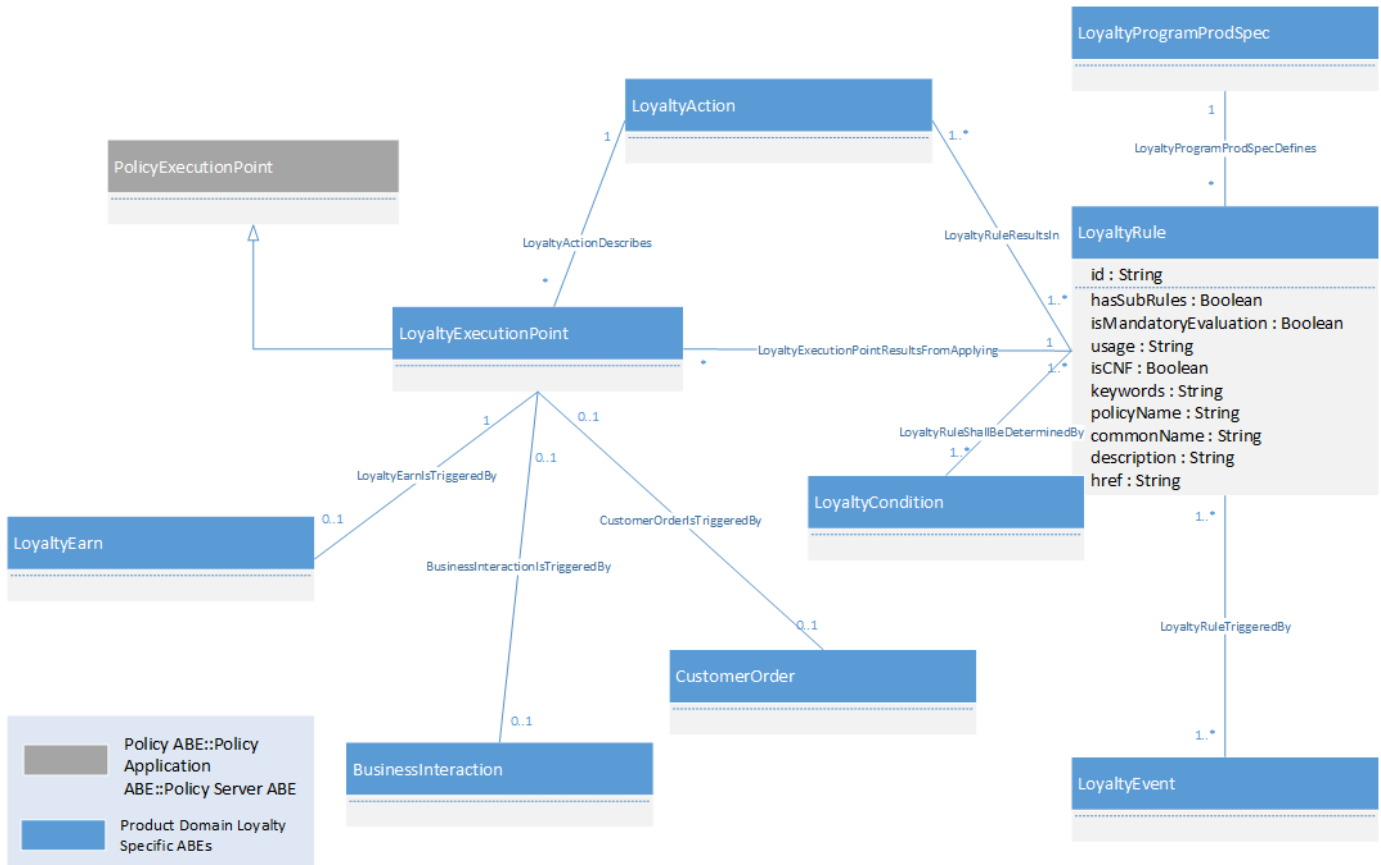


Figure 5 – LoyaltyAction resource model

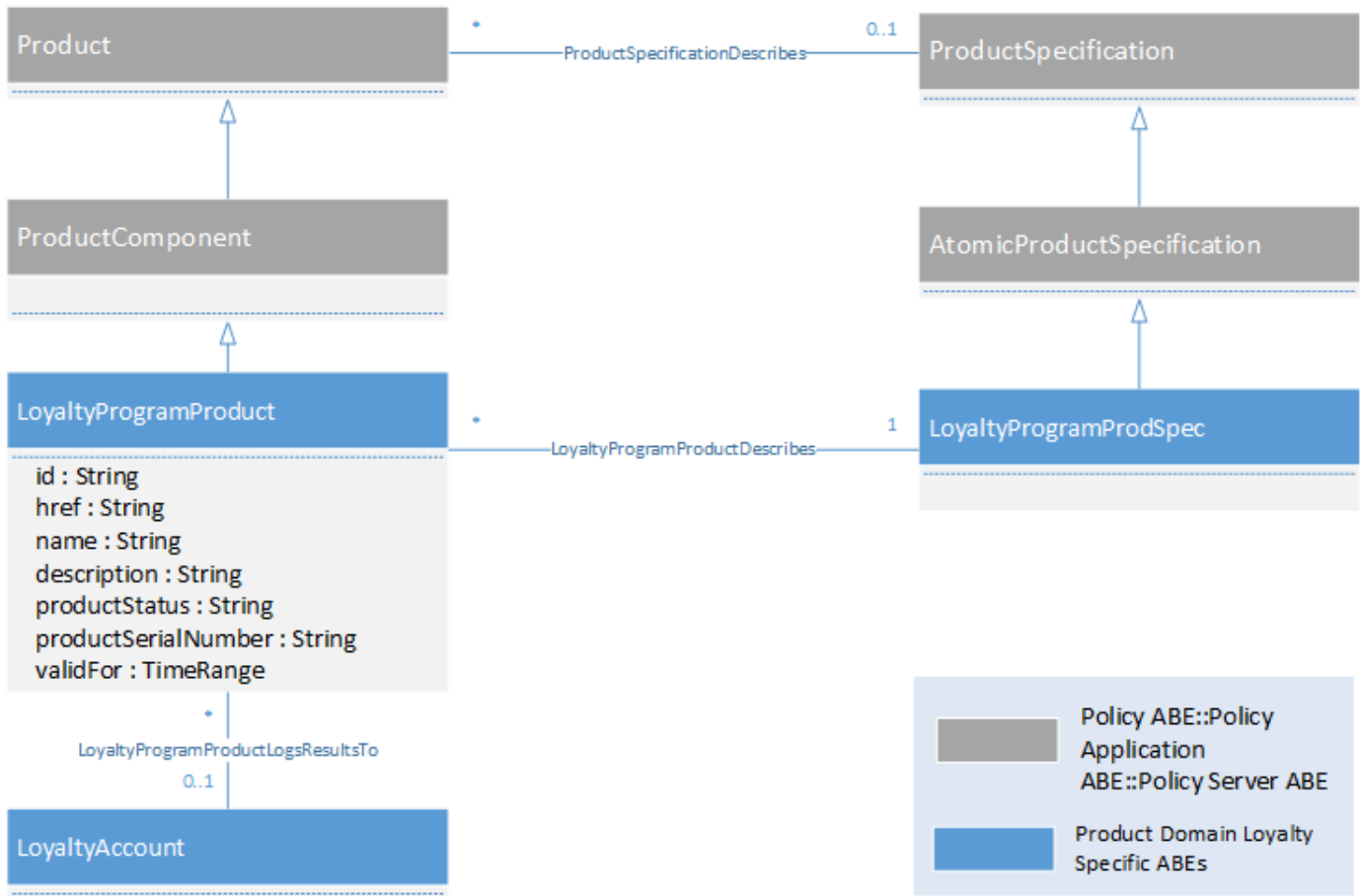


Figure 6– LoyaltyProgramProduct resource model

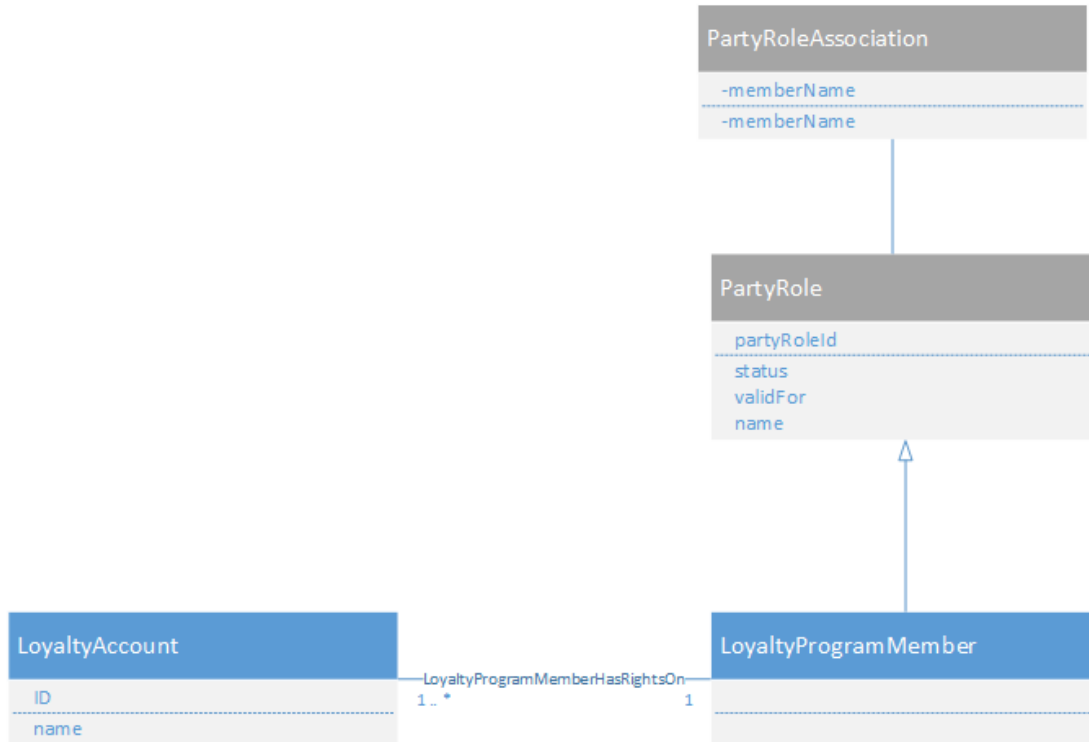
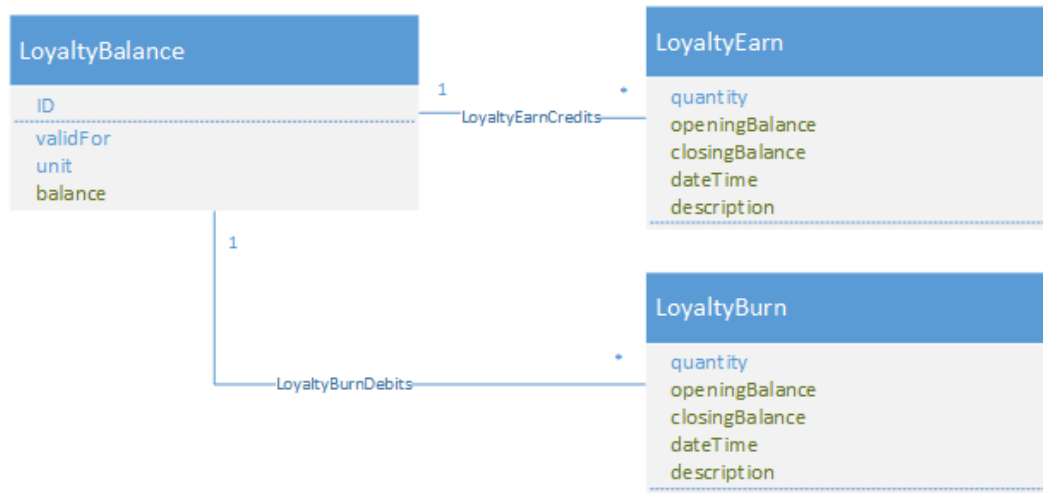


Figure 7 – LoyaltyAccount resource model



Figure 8– LoyaltyBalance resource model



## API OPERATIONS

### OPERATIONS ON PROGRAM PRODUCT SPEC

#### List / Retrieve Program Product Spec

**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 attributes, except href.
- 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.

#### Usage Samples :

REQUEST
GET /loyaltyManagement/loyaltyProgramProductSpec/121 Accept: application/json
RESPONSE
200 Content-Type: application/json { "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": {
  "endTime": "2016-12-31T23:59:59Z",
  "startTime": "2016-01-01T00:00:00Z"
}
}

```

## Patch Program Product Spec

### **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	

**Usage Samples :**

REQUEST
PATCH /loyaltyManagement/loyaltyProgramProductSpec/121 Content-Type: application/json <pre>{   "name": "UpComingProfessionalsProgram",   "description": "Loyalty Program to ensure that prepaid youth market is retained", }</pre>
RESPONSE
201 Content-Type: application/json <pre>{   "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": {     "endTime": "2016-12-31T23:59:59Z",     "startTime": "2016-01-01T00:00:00Z"   } }</pre>

**Create Program Product Spec****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. The created resource may optionally be returned in the response Body.
- 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	

**Usage Samples :****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
(Optional: Created Resource in Response Body)
{
  "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"
  }
}
```

## Delete Program Product Spec

**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.

### Usage Samples :

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

## OPERATIONS ON RULES

### List / Retrieve Rules

#### GET

**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}`.

The LoyaltyRule's subresources (LoyaltyEventType, LoyaltyCondition and LoyaltyAction) links are included in the response body.

### 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.

**Usage Samples :**

REQUEST
GET /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1 Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>{   "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": [{     "id": "8434",     "href": "http://server:port/loyaltyManagement/loyaltyEventType/8424"   }],   {     "id": "4534"     "href": "http://server:port/loyaltyManagement/loyaltyEventType/4534"   } }],</pre>

```

    "loyaltyCondition": [],
    "loyaltyAction": []
  }

```

## Update Rule

### 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

#### Usage Samples :

#### 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 Rule

### 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.

**Usage Samples :**

<b>REQUEST</b>
<pre> PATCH /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/222 Content-Type: application/json {   "isCNF": false,   "isMandatoryEvaluation" : false } </pre>
<b>RESPONSE</b>
<pre> 200 Content-Type: application/json </pre>

```

{
  "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": []
}

```

## Create Rule

### 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. The created resource may optionally be returned in the response Body.
- 404 Not Found – The program spec could not be found.
- 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	false	
isMandatoryEvaluation	N	true	
usage	N		
keywords	N		
policyName	N		
commonName	N		
description	N		

**Usage Samples :**

<b>REQUEST</b>
POST /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule Content-Type: application/json <pre>{   "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" }</pre>
<b>RESPONSE</b>
201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1

*(Optional: Created Resource in Response Body)*

```
{
  "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 Rule

### 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.

#### Usage Samples :

REQUEST
DELETE /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1

Accept: application/json
--------------------------

<b>RESPONSE</b>
-----------------

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

## OPERATIONS ON Rule Conditions

### List / Retrieve Rule Conditions

#### 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 not enabled.
- Attribute selection is not enabled.
- 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.

#### Usage Samples :

<b>REQUEST</b>
----------------

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

<b>RESPONSE</b>
-----------------

```

200
Content-Type: application/json
{
  "id": "2",
  "href": "http://server:port/loyaltyManagement/loyaltyCondition/2"
}
    
```

## Create Rule Condition

### POST

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

#### 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. The created resource may optionally be returned in the response Body.
- 404 Not Found – The program spec or rule could not be found.
- 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

#### Usage Samples :

```

REQUEST

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

<pre>"id" : "2" }</pre>
<b>RESPONSE</b>
<pre>201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyCondition/2 (Optional: Created Resource in Response Body) {   "id": "2",   "href": "http://server:port/loyaltyManagement/loyaltyCondition/2" }</pre>

## Delete Rule Condition

### 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.

#### Usage Samples :

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

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

## OPERATIONS ON Rule Event Types

### List / Retrieve Rule Conditions

#### 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}/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.

#### Usage Samples :

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



## Create Rule Condition

### POST

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

#### 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. The created resource may optionally be returned in the response Body.
- 404 Not Found – The program spec or rule could not be found.
- 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

#### Usage Samples :

REQUEST
POST /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyEventType Content-Type: application/json <pre>{   "id" : "2" }</pre>
RESPONSE
201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyEventType/2

*(Optional: Created Resource in Response Body)*

```
{
  "id": "2",
  "href": "http://server:port/loyaltyManagement/loyaltyEventType/2"
}
```

## Delete Rule Condition

### 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 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.

#### Usage Samples :

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

## OPERATIONS ON Rule Actions

### List / Retrieve Rule Actions

**GET**

**/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyActions/{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.

**Usage Samples :**

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

---

## Create Rule Action

**POST**

**/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyActions**

**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. The created resource may optionally be returned in the response Body.
- 404 Not Found – The program spec or rule could not be found.
- 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

**Usage Samples :**

<b>REQUEST</b>
POST /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyAction Content-Type: application/json <pre>{   "id" : "2" }</pre>
<b>RESPONSE</b>
201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyAction/2 <i>(Optional: Created Resource in Response Body)</i> <pre>{   "id": "2",   "href": "http://server:port/loyaltyManagement/loyaltyAction/2"</pre>

```
}

```

## Delete Rule Action

### DELETE

`/loyaltyManagement/loyaltyProgramProductSpec/{ID}/loyaltyRule/{ID}/loyaltyActions/{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.

#### Usage Samples :

<b>REQUEST</b>
DELETE /loyaltyManagement/loyaltyProgramProductSpec/121/loyaltyRule/1/loyaltyAction/1 Accept: application/json
<b>RESPONSE</b>
200 Content-Type: application/json

## OPERATIONS ON CONDITIONS

### List / Retrieve Conditions

**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 id, attribute and value.
- 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.

**Usage Samples :**

REQUEST
GET /loyaltyManagement/loyaltyCondition/1 Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>{   "id": "1",   "attribute": "age",   "operator": "&lt;",   "value": "30",   "href": "http://server:port/loyaltyManagement/loyaltyCondition/1" }</pre>

---

## Update Conditions

**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

**Usage Samples :**

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

**Patch Condition**

**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	

**Usage Samples :**

REQUEST
PUT /loyaltyManagement/loyaltyCondition/1 Content-Type: application/json <pre>{   "operator": "&lt;=", }</pre>
RESPONSE
200 Content-Type: application/json <pre>{   "id": "1",   "attribute": "age",   "operator": "&lt;=",   "value": "30",   "href": "http://server:port/loyaltyManagement/loyaltyCondition/1" }</pre>



## Create Condition

### 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. The created resource may optionally be returned in the response Body.
- 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		

#### Usage Samples :

**REQUEST**

```
POST /loyaltyManagement/loyaltyCondition
```

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

## RESPONSE

```
201
```

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

```
Content-Location: loyaltyManagement/loyaltyCondition/2
```

```
(Optional: Created Resource in Response Body)
```

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

---

## Delete Condition

### **DELETE** /loyaltyManagement/loyaltyCondition

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.

#### **Usage Samples :**

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

## OPERATIONS ON ACTIONS

### List / Retrieve Actions

#### **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.
- 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.

#### **Usage Samples :**

<b>REQUEST</b>
GET /loyaltyManagement/loyaltyAction/111 Accept: application/json
<b>RESPONSE</b>
200

```
Content-Type: application/json
{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "headers": {
    "Content-Type": "text/json"
  },
  "body": {
    "orderType": "dataBundle",
    "orderCategory": "bundles"
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/111",
  "commonName": "Earn50",
  "description": "",
  "action": "POST",
  "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
              loyaltyBalance/{balanceId}/loyaltyEarn",
  "version": "2.0"
}
```

---

## Update Action

**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

### Usage Samples :

**REQUEST**

PUT /loyaltyManagement/loyaltyAction/2

Content-Type: application/json

```
{
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 80
  },
  "headers": {
    "Content-Type": "text/json"
  },
  "body": {
    "orderType": "dataBundle",
    "orderCategory": "bundles"
  },
  "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",
  "commonName": "Earn80",
  "description": "Award customer loyalty",
  "action": "POST",
}
```

```

"endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
            /loyaltyBalance/{balanceId}/loyaltyEarn",
"version": "5.0"
}

```

## Patch Action

### **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	
version	Y	Should be higher than the previous version.
commonName	Y	
description	Y	
action	Y	

endpoint	Y	
----------	---	--

**Usage Samples :**

REQUEST
PATCH /loyaltyManagement/loyaltyAction/2 Content-Type: application/json <pre>{   "actionAttributes": {     "quantity": 100   },   "version": "6.0" }</pre>
RESPONSE
201 Content-Type: application/json <pre>{   "id": "2",   "type": "LoyaltyEarn",   "actionAttributes": {     "quantity": 100   },   "href": "http://server:port/loyaltyManagement/loyaltyAction/2",   "commonName": "Earn100",   "description": "Award customer loyalty",   "action": "POST",   "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/loyaltyBalance/{balanceId}/loyaltyEarn",   "version": "6.0" }</pre>

**Create Action****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. The created resource may optionally be returned in the response Body.
- 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	N		
body	N		
headers	N		
version	N	1.0	
commonName	N		
description	N		
action	Y		POST, PUT, GET or DELETE
endpoint	Y		

**Usage Samples :**

<b>REQUEST</b>
----------------



POST /loyaltyManagement/loyaltyAction

```
{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "headers": {
    "Content-Type": "text/json"
  },
  "body": {
    "orderType": "dataBundle",
    "orderCategory": "bundles"
  },
  "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

*(Optional: Created Resource in Response Body)*

```
{
  "id": "111",
  "type": "LoyaltyEarn",
  "actionAttributes": {
    "quantity": 50
  },
  "headers": {
    "Content-Type": "text/json"
  },
  "body": {
    "orderType": "dataBundle",
```

```

    "orderCategory": "bundles"
  },
  "href": "http://server:port/loyaltyManagement/loyaltyAction/111",
  "commonName": "LoyaltyEarn",
  "description": "",
  "action": "POST",
  "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
              loyaltyBalance/{balanceId}/loyaltyEarn",
  "version": "2.0"
}

```

---

## Delete Action

### POST /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.

#### Usage Samples :

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

## OPERATIONS ON EVENT TYPES

### List / Retrieve Event Types

#### **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.

#### **Usage Samples :**

REQUEST
GET /loyaltyManagement/loyaltyEventType/111 Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>{   "id": "111",   "eventType": "orderCreationNotification",   "href": "http://server:port/loyaltyManagement/loyaltyEventType/111" }</pre>

### Update Event Type

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

### Usage Samples :

REQUEST
PUT /loyaltyManagement/loyaltyEventType/2 Content-Type: application/json <pre>{   "eventType": "createCustomerInvoice" }</pre>
RESPONSE
201 Content-Type: application/json <pre>{   "id": "2",   "eventType": "createCustomerInvoice",   "href": "http://server:port/loyaltyManagement/loyaltyEventType/2" }</pre>

## Patch Event Type

### 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.
- 409 Conflict – The loyalty event type already exists.
- 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	Must be unique.

**Usage Samples :**

REQUEST
PATCH /loyaltyManagement/loyaltyEventType/2 Content-Type: application/json <pre>{   "eventType": "createCustomerInvoice" }</pre>
RESPONSE
201 Content-Type: application/json <pre>{   "id": "2",   "eventType": "createCustomerInvoice",   "href": "http://server:port/loyaltyManagement/loyaltyEventType/111" }</pre>

**Create Event Type****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. The created resource may optionally be returned in the response Body.
- 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. Must be unique.

**Usage Samples :**

REQUEST
POST /loyaltyManagement/loyaltyEventType <pre>{   "id": "111",   "eventType": "orderCreationNotification" }</pre>
RESPONSE
201 Content-Type: application/json Content-Location: loyaltyManagement/loyaltyEventType/111

```

(Optional: Created Resource in Response Body)
{
  "id": "111",
  "eventType": "orderCreationNotification",
  "href": "http://server:port/loyaltyManagement/loyaltyEventType/111"
}

```

## Delete Event Type

### **DELETE** /loyaltyManagement/loyaltyEventType

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.

#### **Usage Samples :**

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

## OPERATIONS ON LOYALTY PROGRAM PRODUCT

### List / Retrieve Loyalty Program Product

**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.

**Usage Samples :**

REQUEST
GET /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1211 Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>{   "id": "1211",   "name": "DataUsageBenefit",   "description": "Data Usage Loyalty Benefits",   "productStatus": "suspended",   "characteristic": [     {</pre>



```

    "name": "DataUsageThreshold",
    "value": "10"
  },
  {
    "name": "DataUsageThresholdUnitOfMeasure",
    "value": "GB"
  }
],
"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"
}

```

## Patch Loyalty Program Product

### 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).
- If product characteristics are present in the request, it is overwritten. This may still be expanded to support the JSON-PATCH specification.

#### Behavior :

Return status codes

- 200 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	
characteristic	Y	
validFor	Y	
productSpecId	N	
accountId	Y	<p>The loyalty account may be changed, but the same rules apply as with the product creation – if the loyalty program product spec requires a loyaltyAccount and no account ID is specified, an account ID is generated.</p> <p>If a loyalty program product does not require a loyalty account and an ID is specified, an error will be displayed.</p>

#### Usage Samples :

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",
  "characteristic": [
    {
      "name": "DataUsageThreshold",
      "value": "10"
    },
    {
      "name": "DataUsageThresholdUnitOfMeasure",
      "value": "GB"
    }
  ],
  "validFor": {
    "endDateTime": "2017-05-19T16:42:20Z",
    "startDateTime": "2016-04-19T12:00:00Z"
  }
}

```

---

## Create Loyalty Program Product

**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. The created resource may optionally be returned in the response Body.
- 409 Conflict – Resource already exists.
- 422 Unprocessable Entity – There was a field rule violation or missing required attribute.

Attribute name	Man dato ry	Default	Rule
id	N	Automaticall y 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	
validFor	N		
productSpecId	Y		Should be an existing LoyaltyProgramProductSpecification
accountId	N		<p>Specify an existing account ID to be used.</p> <p>If the LoyaltyProgramProductSpec's needsLoyaltyAccount attribute is TRUE and both the accountId and loyaltyAccount attribute was omitted, then an error is returned.</p> <p>If the accountId attribute was provided and the LoyaltyProgramProductSpec's needsLoyaltyAccount attribute is FALSE, an error is returned.</p> <p>If the accountId attribute was provided and the</p>

			LoyaltyProgramProductSpec's needsLoyaltyAccount attribute is TRUE, it must correspond to one of the LoyaltyMember's existing loyaltyAccounts.
loyaltyAccount	N		If the account does not yet exist, the account details is specified in the loyaltyAccount section.
loyaltyAccount.id	N		The new account ID is autogenerated if it is not present.
loyaltyAccount.loyaltyBalance	Y		A new account must have at least one balance.
loyaltyAccount.loyaltyBalance.quantity.unit	Y		
loyaltyAccount.loyaltyBalance.quantity.balance	N	0.00	
loyaltyAccount.loyaltyBalance.validFor.startDate Time	N		
loyaltyAccount.loyaltyBalance.validFor.endDate Time	N		

**Usage Samples :**

The example below shows creating a new product with an existing account.

<b>REQUEST</b>
<pre> POST /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct Content-Type: application/json {   "id": "1213",   "name": "PrepaidTopupBenefits",   "description": "Prepaid account top-up benefits",   "productStatus": "active",   "characteristic": [     {       "name": "DataUsageThreshold", </pre>

```

    "value": "10"
  },
  {
    "name": "DataUsageThresholdUnitOfMeasure",
    "value": "GB"
  }
],
"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

*(Optional: Created Resource in Response Body)*

```

{
  "id": "1213",
  "name": "PrepaidTopupBenefits",
  "description": "Prepaid account top-up benefits",
  "productStatus": "active",
  "characteristic": [
    {
      "name": "DataUsageThreshold",
      "value": "10"
    },
    {
      "name": "DataUsageThresholdUnitOfMeasure",
      "value": "GB"
    }
  ],
  "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"
}
}
```

The example below shows creating a new product with a new account.

## REQUEST

POST /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct

Content-Type: application/json

```
{
  "id": "1213",
  "name": "PrepaidTopupBenefits",
  "description": "Prepaid account top-up benefits",
  "productStatus": "active",
  "characteristic": [
    {
      "name": "DataUsageThreshold",
      "value": "10"
    },
    {
      "name": "DataUsageThresholdUnitOfMeasure",
      "value": "GB"
    }
  ],
  "validFor": {
    "endDateTime": "2017-05-19T16:42:20Z",
    "startDateTime": "2016-04-19T12:00:00Z"
  },
  "productSpecId": "1",
  "loyaltyAccount": {
    "id": "PaulLoyalty",
    "loyaltyBalance": {
      "quantity": {
        "unit": "points",
        "balance": "10.00"
      },
      "validFor": {
        "startDateTime": "2017-05-19T16:42:20Z",
        "endDateTime": "2018-04-19T12:00:00Z"
      }
    }
  }
}
```



**RESPONSE**

201

Content-Type: application/json

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

*(Optional: Created Resource in Response Body)*

```

{
  "id": "1213",
  "name": "PrepaidTopupBenefits",
  "description": "Prepaid account top-up benefits",
  "productStatus": "active",
  "characteristic": [
    {
      "name": "DataUsageThreshold",
      "value": "10"
    },
    {
      "name": "DataUsageThresholdUnitOfMeasure",
      "value": "GB"
    }
  ],
  "loyaltyProgramProductSpec": {
    "id": "121",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramProductSpec/121"
  },
  "loyaltyAccount": {
    "id": "PaulLoyalty",
    "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/121/loyaltyAccount/PaulLoyalty"
  },
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1213",
  "validFor": {
    "endDateTime": "2017-05-19T16:42:20Z",
    "startDateTime": "2016-04-19T12:00:00Z"
  }
}

```

**Delete Loyalty Program Product**

**DELETE**

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

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

If the LoyaltyAccount associated with the LoyaltyProgramProduct is used only on the addressed loyaltyProgramProduct, the associated LoyaltyAccount will also be deleted. However, if the LoyaltyAccount is reused on another LoyaltyProgramProduct, the LoyaltyAccount will not be deleted.

**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.

**Usage Samples :**

<b>REQUEST</b>
DELETE /loyaltyManagement/loyaltyProgramMember/121/loyaltyProgramProduct/1213 Accept: application/json
<b>RESPONSE</b>
200 Content-Type: application/json

## OPERATIONS ON LOYALTY EXECUTION POINT

### List / Retrieve Loyalty Execution Points

**GET**

**/loyaltyManagement/loyaltyProgramMember/{ID}/loyaltyProgramProduct/{ID}/loyaltyExecutionPoint/{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/loyaltyProgramProduct/{ID}/loyaltyExecutionPoint with no {ID}.

**Description :**

- This operation is used to retrieve the loyalty execution points for a loyalty program product.
- Filtering selection is enabled on all first level attributes.
- Attribute selection is enabled on all first level 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 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.

**Usage Samples :**

REQUEST
GET /loyaltyManagement/loyaltyProgramMember/348243/loyaltyProgramProduct/1211/loyaltyExecutionPoint Accept: application/json
RESPONSE
200 Content-Type: application/json <pre>[{   "id": "2",   "type": "LoyaltyEarn",   "actionAttributes": {     "quantity": 50   },   "href":   "http://server:port/loyaltyManagement/loyaltyProgramMember/348243/loyaltyProgramProduct/1211/loyaltyExecutionPoint/2",   "commonName": "Earn50",   "description": "",   "action": "POST",   "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/loyaltyBalance/{balanceId}/loyaltyEarn",   "version": "2.0",   "dateTime": "2017-05-19T16:42:20Z " },</pre>

```
{
  "id": "434",
  "type": "BusinessInteraction",
  "actionAttributes": {
    "quantity": 50
  },
  "href":
  "http://server:port/loyaltyManagement/loyaltyProgramMember/348243/loyaltyProgramProduct/1211/loyalty
  ExecutionPoint/434",
  "commonName": "Earn50",
  "description": "",
  "action": "POST",
  "endpoint": "http://server:port/loyaltyManagement/loyaltyProgramMember/{memberId}/
  loyaltyBalance/{balanceId}/loyaltyEarn",
  "version": "2.0",
  "dateTime": "2016-05-19T16:42:20Z "
}
```

## OPERATIONS ON LOYALTY PROGRAM MEMBER

### List / Retrieve Loyalty Program Members

#### **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 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.

#### **Usage Samples :**

<b>REQUEST</b>
GET /loyaltyManagement/loyaltyProgramMember/JDSU778DS Accept: application/json
<b>RESPONSE</b>
200 Content-Type: application/json  <pre>{   "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"   } }</pre>

## Update Loyalty Program Members

**PUT /loyaltyManagement/loyaltyProgramMember / {ID}**

This Uniform Contract operation is used to update the entire 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.

### 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.

### Usage Samples :

<b>REQUEST</b>
PATCH /loyaltyManagement/loyaltyProgramMember/JDSU778DS 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

```
{
  "id": "JDSU778DS",
  "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS",
  "status": "suspended",
  "name": "James Joe",
  "validFor": {
    "startDateTime": "2015-04-19T16:42:23.0Z ",
    "endDateTime": "2016-04-19T16:42:23.0Z"
  }
}
```

---

## Patch Loyalty Program Members

**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.

**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.

**Usage Samples :**

REQUEST
PATCH /loyaltyManagement/loyaltyProgramMember/JDSU778DS Content-type: application/json  <pre>{   "status": "suspended",   "name": "James Joe" }</pre>
RESPONSE
201 Content-Type: application/json  <pre>{   "id": "JDSU778DS",   "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/JDSU778DS",   "status": "suspended",   "name": "James Joe",   "validFor": {     "startDateTime": "2015-04-19T16:42:23.OZ ",     "endDateTime": "2016-04-19T16:42:23.OZ"   } }</pre>

## Create Loyalty Program Members

### 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. The created resource may optionally be returned in the response Body.
- 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
(Optional: Created Resource in Response Body)

{
  "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"
}
```

---

## Delete Loyalty Program Members

**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.

### Usage Samples :

<b>REQUEST</b>
DELETE /loyaltyManagement/loyaltyProgramMember/JDSU778DS
<b>RESPONSE</b>
200

## OPERATIONS ON LOYALTY ACCOUNT

### List / Retrieve Loyalty Balances

**GET /loyaltyManagement/loyaltyAccount/{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}

Only GET operations are supported under the LoyaltyAccount resource, as LoyaltyAccounts are linked to a LoyaltyProgramProduct and cannot be modified without referencing the LoyaltyAccount.

#### 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 productId only.
- 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.

#### Usage Samples :

<b>REQUEST</b>
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"
  }
}]

```

**OPERATIONS ON LOYALTY BALANCE****List / Retrieve Loyalty Balances****GET /loyaltyManagement/loyaltyAccount/{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/loyaltyAccount/{ID}/loyaltyBalance with no {ID}.

**Description :**

- This operation retrieves loyalty program member account balances.
- 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 not enabled on balances.
- 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 account or loyalty balance could not be found.

**Usage Samples :****REQUEST**

GET /loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance  
 Accept: application/json

## RESPONSE

200

Content-Type: application/json

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

## Patch Loyalty Balances

### PATCH

**/loyaltyManagement/loyaltyAccount/{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 account 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 account or loyalty balance could not be found.
- 422 Unprocessable Entity – There was a field rule violation.

Attribute name	Patchable	Rule
id	N	
quantity.unit	N	
quantity.balance	N	
validFor.startDateTime	Y	
validFor.endDateTime	Y	The end date must be after the start date.
loyaltyProgramMember	N	Managed on the loyalty program member endpoint.

**Usage Samples :**

REQUEST
PATCH /loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/iTunes Content-type: application/json  <pre>{   "validFor": {     "startDateTime": "2016/02/19 18:42:23",     "endDateTime": "2018/12/30 17:42:23"   } }</pre>
RESPONSE
201 Content-Type: application/json  <pre>{   "id": "iTunes",   "href": "/loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/iTunes",   "quantity": {     "unit": "NZD",</pre>

```

    "balance": 300
  },
  "loyaltyProgramMember": {
    "id": "PHDUIU8336",
    "href": "/loyaltyManagement/loyaltyProgramMember/PHDUIU8336 "
  }
  "validFor": {
    "startDateTime": "2016/02/19 18:42:23",
    "endDateTime": "2018/12/30 17:42:23"
  }
}

```

## Post Loyalty Balances

### POST

#### /loyaltyManagement/loyaltyAccount/{ID}/loyaltyBalance

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

#### Description :

- This operation creates a new loyalty 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. The created resource may optionally be returned in the response Body.
- 404 – The loyalty account could not be found.
- 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.	
quantity.unit	Y		

quantity.balance	N	0.00	
validFor.startDateTime	N		
validFor:endDateTime	N		Must be after the start date.

**Usage Samples :**

<b>REQUEST</b>
<p>POST /loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance Content-type: application/json</p> <pre>{   "id": "839",   "unit": "points" }</pre>
<b>RESPONSE</b>
<p>201 Content-Type: application/json Content-Location: /loyaltyManagement/loyaltyProgramMember/PHDUIU8336/loyaltyBalance/839 (Optional: Created Resource in Response Body)</p> <pre>{   "id": "839",   "href": "http://server:port/loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/839",   "quantity": {     "unit": "NZD",     "balance": 300   },   "validFor": {     "startDateTime": "2016-07-08 17:15:41"   },   "loyaltyProgramMember": {     "id": "PHDUIU8336",     "href": "http://server:port/loyaltyManagement/loyaltyProgramMember/PHDUIU8336 "   } }</pre>

## Delete Loyalty Balances

### DELETE

**/loyaltyManagement/loyaltyAccount/{ID}/loyaltyBalance/{ID}**

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

#### Description :

- This operation deletes a loyalty account balance.

#### Behavior :

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

#### Usage Samples :

<b>REQUEST</b>
DELETE /loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/839
<b>RESPONSE</b>
200

## OPERATIONS ON LOYALTY EVENT

### List / Retrieve Loyalty Earn

#### GET

**/loyaltyManagement/loyaltyAccount/{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/loyaltyAccount/{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 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 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 account, loyalty balance or loyalty earn transaction could not be found.

### Usage Samples :

REQUEST
GET /loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/iTunes/loyaltyEarn Accept: application/json
RESPONSE
200 Content-Type: application/json  <pre> [[   {     "id": "843G-838F-HY23-0238",     "href": "http://server:port/loyaltyManagement/loyaltyAccount/JohnLoyalty/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/loyaltyAccount/JohnLoyalty/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."   } ]]           </pre>

## Create Loyalty Earn

### POST

`/loyaltyManagement/loyaltyAccount/{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. The created resource may optionally be returned in the response Body.
- 404 Not Found – The account or balance could not be found.
- 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	

#### Usage Samples :

REQUEST
<pre>POST /loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/iTunes/loyaltyEarn Content-type: application/json {</pre>

```

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

```

## RESPONSE

```

201
Content-Type: application/json
Content-Location:
/loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/iTunes/loyaltyEarn/738F-039J-2636-LDH8

```

*(Optional: Created Resource in Response Body)*

```

{
  "id": "738F-039J-2636-LDH8",
  "href": "http://server:port/loyaltyManagement/loyaltyAccount/JohnLoyalty/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."
}

```

## OPERATIONS ON LOYALTY BURN

### List / Retrieve Loyalty Burn

#### GET

**/loyaltyManagement/loyaltyAccount/{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/loyaltyAccount/{ID}/loyaltyBalance/{ID}/loyaltyBurn with no {ID}

#### Description :

- This operation retrieves a loyalty program 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 account, loyalty balance or loyalty earn transaction could not be found.

**Usage Samples :**

REQUEST
GET /loyaltyManagement/loyaltyAccount/JohnLoyalty/loyaltyBalance/iTunes/loyaltyBurn Accept: application/json
RESPONSE
200 Content-Type: application/json  <pre> [   {     "id": "94JU-03J8-57S4-0893",     "href": "http://server:port/loyaltyManagement/loyaltyAccount/JohnLoyalty/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/loyaltyAccount/JohnLoyalty/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>

## Create Loyalty Burn

### POST

`/loyaltyManagement/loyaltyAccount/{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. The created resource may optionally be returned in the response Body.
- 404 Not Found – The account or balance could not be found.
- 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	

#### Usage Samples :

REQUEST
POST /loyaltyManagement/loyaltyAccount/JohanLoyalty/loyaltyBalance/iTunes/loyaltyBurn Content-type: application/json

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

**RESPONSE**

```
201
Content-Type: application/json
Content-Location: /loyaltyManagement/loyaltyAccount/JohanLoyalty/loyaltyBalance/
iTunes/loyaltyBurn/738F-039J-2636-LDH8
(Optional: Created Resource in Response Body)

{
  "id": "738F-039J-2636-LDH8",
  "href": "http://server:port/loyaltyManagement/loyaltyAccount/JohanLoyalty/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."
}
```

**OPERATIONS ON LOYALTY EVENT****Create Loyalty Event**

**GET** /loyaltyManagement/loyaltyEvent/{ID}

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. The created resource may optionally be returned in the response Body.
- 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		
memberId	Y		

#### Usage Samples :

REQUEST
POST /loyaltyManagement/loyaltyEvent Content-type: application/json <pre>{   "eventId": "00001",   "eventTime": "2013-04-19T16:42:25-04:00",   "eventType": "orderCreationNotification",   "memberId": "11843385",   "event": {     "productOrder": {       "id": "42",       "href": "http://serverlocation:port/orderManagement/productOrder/42",       "externalId": "NiceNameForTheConsumer_42",       ...     }   } }</pre>
RESPONSE
201 Content-Type: application/json

## API NOTIFICATION TEMPLATES

### NOTIFICATIONS ON LOYALTY PROGRAM MEMBER

#### Register Loyalty Program Member 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.

**Usage Samples :**

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

#### Unregister Loyalty Program Member 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.



**Usage Samples :**

<b>REQUEST</b>
DELETE /loyaltyManagement/loyaltyProgramMember/hub/42 Accept: application/json
<b>RESPONSE</b>
204

---

**Publish Loyalty Program Member Listener**
**POST /client/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.

**Usage Samples :**

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

201  
 Content-Type: application/json

## NOTIFICATIONS ON LOYALTY MEMBER PRODUCT

### Register Loyalty Member Product 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.

#### Usage Samples :

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 Loyalty Member Product 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.

#### Usage Samples :

<b>REQUEST</b>
DELETE /loyaltyManagement/loyaltyProgramMemberProduct/hub/42 Accept: application/json
<b>RESPONSE</b>
204

---

## Publish Loyalty Member Product Listener

### POST /client/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.

#### Usage Samples :

<b>REQUEST</b>
POST /client/listener Accept: application/json
{  "eventId": "111", "eventType": "LoyaltyProgramMemberProductCreationNotification", "event": {

<pre> EVENT BODY as described in loyaltyProgramProductMember event model section. } } </pre>
<b>RESPONSE</b>
<pre> 201 Content-Type: application/json </pre>

## NOTIFICATIONS ON LOYALTY EARN

### Register Loyalty Earn 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.

#### Usage Samples :

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

### Unregister Loyalty Earn 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.

### Usage Samples :

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

---

## Publish Loyalty Earn Listener

### POST /client/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.

### Sample Usage :

REQUEST
POST /client/listener Accept: application/json  { "eventId": "111", "eventType": "LoyaltyEarnNotification", "event": {

<pre> EVENT BODY as described in loyaltyEarn event model section.   } }</pre>
<b>RESPONSE</b>
<pre> 201 Content-Type: application/json</pre>

## NOTIFICATIONS ON LOYALTY BURN

### Register Loyalty Burn 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.

##### **Sample Usage :**

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

### Unregister Loyalty Burn 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.

### **Usage Samples :**

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

---

## Publish Loyalty Burn Listener

### **POST /client/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.

### **Usage Sampels :**

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

## NOTIFICATIONS ON LOYALTY EVENT

### Register Loyalty Event 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.

**Usage Samples :****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 Loyalty Event 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.

### Usage Samples :

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

---

## Publish Loyalty Event Listener

### POST /client/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.

### Sample Usages :

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

## Event Models

### Loyalty Program Member Notification Model

#### 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"
      }
    }
  }
}

```

Field	Description
<b>eventType</b>	The event type. This value is LoyaltyProgramMemberCreateNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyProgramMember</b>	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"
    }
  }
}
}
}
}

```

Field	Description
<b>eventType</b>	The event type. This value is LoyaltyProgramMemberUpdateNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyProgramMember</b>	The loyaltyProgramMember that was created.

## loyaltyProgramMemberDeleteNotification

```

{
  "eventType": "LoyaltyProgramMemberDeleteNotification",
  "eventTime": "2015-10-28T05:46:25.0Z",
  "eventId": "3958",
  "event": {
    "loyaltyProgramMember": {
      "status": "active",
      "name": "James Joe",
      "validFor": {
        "startDateTime": "2015-04-19T16:42:23.0Z ",
        "endDateTime": "2016-04-19T16:42:23.0Z"
      }
    }
  }
}
}
}
}

```

Field	Description
<b>eventType</b>	The event type. This value is LoyaltyProgramMemberDeleteNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyProgramMember</b>	The details of the loyaltyProgramMember before it was deleted.

## Loyalty Program Member Product Model

### loyaltyProgramMemberProductCreateNotification

```

{
  "eventType": "LoyaltyProgramMemberProductCreationNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "4343",
  "event": {
    "loyaltyProgramProduct": {
      "id": "1211",
      "name": "DataUsageBenefit",
      "description": "Data Usage Loyalty Benefits",
      "productStatus": "suspended",
      "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
<b>eventType</b>	The event type. This value is LoyaltyProgramMemberProductCreateNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyProgramProduct</b>	The loyaltyProgramProduct that was created.

## loyaltyProgramMemberUpdateNotification

```

{
  "eventType": "LoyaltyProgramMemberProductUpdateNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "4343",
  "event": {
    "loyaltyProgramProduct": {
      "id": "1211",
      "name": "DataUsageBenefit",
      "description": "Data Usage Loyalty Benefits",
      "productStatus": "active",
      "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
<b>eventType</b>	The event type. This value is LoyaltyProgramMemberProductUpdateNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyProgramProduct</b>	The loyaltyProgramProduct that was updated.

## loyaltyProgramMemberDeleteNotification

```

{
  "eventType": "LoyaltyProgramMemberProductDeleteNotification",
  "eventTime": "2015-09-27T05:46:25.0Z",
  "eventId": "9438",
  "event": {
    "loyaltyProgramProduct": {
      "id": "1211",
      "name": "DataUsageBenefit",
      "description": "Data Usage Loyalty Benefits",
      "productStatus": "active",
      "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
<b>eventType</b>	The event type. This value is LoyaltyProgramMemberProductDeleteNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyProgramProduct</b>	The loyaltyProgramProduct before it was deleted.

## Loyalty Earn Model

### 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
<b>eventType</b>	The event type. This value is LoyaltyEarnNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyEarn</b>	The loyaltyEarn that was transacted.

## Loyalty Burn Model

### 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
<b>eventType</b>	The event type. This value is LoyaltyBurnNotification for this event model.
<b>eventId</b>	A unique event identifier.
<b>event.loyaltyBurn</b>	The loyaltyBurn that was transacted.

## Loyalty Event Model

### loyaltyEventNotification

```

{
  "eventType": "LoyaltyEarnNotification",
  "eventTime": "2016-06-04T07:42:25.0Z",
  "eventId": "3983",
  "memberId": "4893294389",
  "event": {
    "loyaltyEvent": {
      All the incoming event attributes.
    }
  }
}
}

```

Field	Description
<b>eventType</b>	The event type. This value is LoyaltyEvent for this event model.
<b>eventTime</b>	The time that the event occurred.
<b>eventId</b>	A unique event identifier.

---

<b>memberId</b>	The loyalty program member ID associated with the event.
<b>event.loyaltyEvent</b>	All the attributes incoming from the LoyaltyEvent.

---

## 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**

<b>Version Number</b>	<b>Date</b>	<b>Release led by:</b>	<b>Description</b>
Release 1.0	04/15/2013	Clara van Staden	First Release of Draft Version of the Document.
Release 1.1		Alicja Kawecki	Updated cover and Notice to reflect TM Forum Team Approved status.
Release 2.0.0	07/19/2017	API Team	Updates for Fx17
Release 17.0.1 Version 2.0.1	20-Nov-2017	Adrienne Walcott	Updated to reflect TM Forum Approved Status

## COMPANY CONTACT DETAILS

Company	Team Member Representative
Globetom	Name: Clara van Staden Title: Senior Software Engineer Email: <a href="mailto:clara.vstaden@globetom.com">clara.vstaden@globetom.com</a> Phone: +27 84 206 4826