Release Notes: Release Ready

SabreSonic CSS Digital Connect Check-In REST Services

Version 3.1.7

Software Version 3.1.7

Document Edition 1.0 (December 2020)

Table of Contents

1 Introduction

2 Release Features

3 Defect Fixes

1 Introduction

1.1 Document Overview

This document contains release notes information for *SabreSonic CSS Digital Connect Check-In REST Services*® Release 3.1.7. Read this document so that you are aware of changes to the solution.

1.2 Release Identification

Release Version	Type (Version, Update, or Patch)	Date	Approved By	Description of Change
3.1.7	Version	1 December 2020	Grzegorz Kurek, Rafal Holub	Software updated

2

Release Features

2.1 Summary of Features

Digital Connect Check-In is introducing new features and enhancements in the following areas:

- Recognizing Frequent Flyer Program Members of Partner Airlines
- Optimization Process for Baggage Pooling
- API-first Design with OpenAPI Specification

The following sections explain more details about the new items.

2.1.1 Recognizing Frequent Flyer Program Members of Partner Airlines

For Passengers and Agent facing applications *Digital Connect Check-In* introduces possibility to recognize Frequent Flyer program members not only for hosting Airlines but also for the partner Airlines.

Therefore, *Digital Connect Check-In* returns more details about Frequent Flyer program in the /dcci/passenger/details service.

Prerequisites

None

Limitations

None

Configuration

Property Name	Description	Default Value	Required Value
s4ci.feature.passengerDetails.returnPnrLoyaltyAccount	Determines whether the details from a PNR regarding the loyalty account should be populated or not.	false	true

2.1.1.1 Returning Partner Frequent Flyer Traveler Suite (FQTV) Details in the /dcci/passenger/details Service

When *Digital Connect Check-In* retrieves the reservation, then some details about the loyalty program are also returned, which allows POS to recognize the passengers with the higher level and enable some additional features or options for them.

In the past *Digital Connect Check-In* did not return the banner details which are required by POS for correct handling.

Now *Digital Connect Check-In* enriches the Loyalty data with additional details available in the reservation. The following information about the FQTV details are returned according to the tier level:

- Tier level priority
- ATPCO tier level
- Customer Insight tier level.

This information might be used to compare the different tiers of hierarchy.

Prerequisites

None

Limitations

None

Configuration

Property Name	Description	Default Value	Required Value
s4ci.feature.passengerDetails.returnPnrLoyaltyAccount	Determines whether the details from a PNR regarding the loyalty account should be populated or not.	false	true

2.1.1.2 Returning Frequent Flyer Traveler Suite (FQTV) Details after Adding FQTV Number to the Reservation

When POS adds FQTV details through the /dcci/passenger/update service, the details of already added account are available.

Prerequisites

None

Limitations

The feature is available only for the path where PNR Update Reservation is used to add FQTV details.

Configuration

Property Name	Description	Default Value	Required Value
s4ci.feature.passengerDetails.returnPnrLoyaltyAccount	Determines whether the details from a PNR regarding the loyalty account should be populated or not.	false	true

Property Name	Description	Default Value	Required Value
se.s4ci.updatereservation.frequentFlierValidationEnabled	Specifies whether the validation of the Frequent Flier account is enabled. By default, it is 'false', which results in a call to the airport system. If this flag is 'true', then the reservation system with an additional validation is called.	false	true

2.1.2 Optimization Process for Baggage Pooling

Baggage Pool functionality allows to share baggage allowance between the passengers as well as pre-paids and charges.

Therefore, *Digital Connect Check-In* introduces optimization processes related to the calculating baggage charge, booking charges and issuing bag tags.

Prerequisites

None

Limitations

None

Configuration

Property Name	Description	Default Value	Required Value
s4ci.feature.baggage.book.optimization.enabled	Determines whether the /dcci/baggage/book service should optimize the downline calls.	false	true
s4ci.feature.baggage.book.enhancedResults.enabled	Determines whether the /dcci/baggage/book service returns the enhanced results.	false	true
s4ci.feature.passengerDetails.optimization.enabled	Determines whether the calls to the downline service should be optimized or not.	false	true

2.1.2.1 The /dcci/baggage/book Service Optimization

The /dcci/baggage/book service can be used to book charges required after the baggage calculations triggered by using the /dcci/baggage/price.

So far *Digital Connect Check-In* has created AirExtras one by one. With the optimization all requested charges are booked at once by using a single operation.

This optimization includes also the enhancement of the /dcci/baggage/book response, and the results of the requested transaction are more detailed including the information about used pre-paid AirExtras and linked charges.

Prerequisites

None

Limitations

None

Configuration

Property Name	Description	Default Value	Required Value
s4ci.feature.baggage.book.optimization.enabled	Determines whether /baggage/book should optimize downline calls.	false	true
s4ci.feature.baggage.book.enhancedResults.enabled	Determines whether /baggage/book returns enhanced results.	false	true

2.1.2.2 Connecting Segments in the /dcci/passenger/details Service

When *Digital Connect Check-In* retrieves passenger details by using the downline service, a call to this service needs to be performed for each connecting segment. To improve the performance of retrieving necessary passenger's data, the downline service introduces the possibility to retrieve all connecting segments using a single call to their service.

Therefore, *Digital Connect Check-In* consumes this new functionality of the downline service to perform less calls and improve performance of the transactions.

Prerequisites

None

Limitations

None

Configuration

Property Name	Description	Default Value	Required Value
s4ci.feature.passengerDetails.optimization.enabled	Determines whether the calls to the downline service should be optimized or not.	false	true

2.1.3 API-first Design with OpenAPI Specification

Digital Connect Check-In 3.1.7 introduces API-first design with an OpenAPI Specification. This allows to better address market needs and deliver APIs faster.

All existing specifications were reviewed, reworked and adjusted (where applicable), which resulted in delivering OpenAPI Specification that is in line with the current *Digital Connect Check-In* APIs.

There were no changes in the services paths, HTTP methods and data models. All *Digital Connect Check-In* APIs remain backward compatible and no action is required from the service consumers to adjust to these changes.

The initiative focused solely on delivering the highest quality OpenAPI-compatible API specification for all existing and future *Digital Connect Check-In* REST services.

During the analysis it was found that some of the model definitions in the previous *Digital Connect Check-In* API specifications contained invalid property names. The problem was specific only to the API specification as the actual API data model did not change since the initial release.

The following is the list of the changes:

Definition Name	Previous Property Name	Data Model Property Name
res.AirExtra	emdSegmentCoupons	eMDSegmentCoupons
res.EMDSegmentCoupons	emdSegmentCoupon	eMDSegmentCoupon
res.ResultCodes	avs	aVS
res.ResultCodes	csc	cSC
res.ResultCodes	cavv	cAVV
ress.PassengerEmailRef	passengerEmailRef	value

Definition Names and References

Additionally, the definition names and references were adjusted to comply with RFC 3986. The change was limited to a substitution of the colon character (:) with a dot character (.).

Example:

Definition res:AirExtra was renamed to res.AirExtra.

All references ($ref) were adjusted in the same way. There were no changes to case or naming.

Data Model Composition

Last change was related to the data model composition. In the previous API specifications, all models were 'flat' and contained multiple redundant fields and documentation entries. Now they are remodeled by using Swagger v2 “allOf” property.

Example (simplified):

Before

```
{
  "prc:AirExtra": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string"
      },
      "syntheticIdentifier": {
        "type": "string"
      },
      "ancillary": {
        "$ref": "#/definitions/anc:Ancillary"
      },
      "miscFee": {
        "$ref": "#/definitions/fee:MiscFee"
      },
      "paymentStatus": {
        "$ref": "#/definitions/sec:AirExtraPaymentStatus"
      },
      "pnrId": {
        "type": "string"
      }
    }
  },
  "prc:DetailedAirExtra": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string"
      },
      "syntheticIdentifier": {
        "type": "string"
      },
      "ancillary": {
        "$ref": "#/definitions/anc:Ancillary"
      },
      "miscFee": {
        "$ref": "#/definitions/fee:MiscFee"
      },
      "paymentStatus": {
        "$ref": "#/definitions/sec:AirExtraPaymentStatus"
      },
      "pnrId": {
        "type": "string"
      },
      "quantity": {
        "type": "integer",
        "format": "int32"
      },
      "segmentRefs": {
        "$ref": "#/definitions/sec:SegmentRefs"
      },
      "unitFee": {
        "$ref": "#/definitions/anc:TotalFee"
      },
      "totalFee": {
        "$ref": "#/definitions/anc:TotalFee"
      }
    }
  }
}
```

Now

```
{
  "prc.AirExtra": {
    "type": "object",
    "properties": {
      "ancillary": {
        "$ref": "#/definitions/anc.Ancillary"
      },
      "id": {
        "type": "string"
      },
      "syntheticIdentifier": {
        "type": "string"
      },
      "miscFee": {
        "$ref": "#/definitions/fee.MiscFee"
      },
      "paymentStatus": {
        "$ref": "#/definitions/sec.AirExtraPaymentStatus"
      },
      "pnrId": {
        "type": "string"
      }
    }
  },
  "prc.DetailedAirExtra": {
    "allOf": [
      {
        "$ref": "#/definitions/prc.AirExtra"
      },
      {
        "type": "object",
        "properties": {
          "quantity": {
            "type": "integer",
            "format": "int32"
          },
          "segmentRefs": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/sec.Segment"
            }
          },
          "unitFee": {
            "$ref": "#/definitions/anc.TotalFee"
          },
          "totalFee": {
            "$ref": "#/definitions/anc.TotalFee"
          }
```

```
                }
            }
        ]
    }
}
```

Note By default *Digital Connect Check-In* API Specifications are released as OpenAPI Specification 2.0 (known as Swagger v2) with 'allOf'-based composition. On demand they can be also delivered as OpenAPI Specification 3.0.x with the disabled composition in both JSON and YAML formats.

• • •

3

Defect Fixes

3.1 Summary of Defects

The following *Digital Connect Check-In* defect has been fixed for this release.

3.1.1 Wrong results of the eligibility recalculation when done before the refreshing passenger's ancillaries

Customer Tracking #: N/A	**Sabre Tracking #:** CSSDC-1728
Description: Eligibility recalculation in the cancel / delete services ended up with a wrong result because it was triggered before the ancillaries data refresh.	
Resolution: It has been fixed by changing the order of the actions in the Update Reservation flow. Refresh of ancillaries is now triggered before the eligibility recalculation action.	
Programs/Modules Affected: /dcci/seats/cancel /dcci/ancillaries/cancel /dcci/baggage/cancel	