App Connect Test Cases and Results

# Malformed Data 1

|  |  |
| --- | --- |
| *Test Case* | Push malformed CSV Data inside the JSON Data to App Connect using API call. Clubbed first 2 rows (removed newline between row 1 and row 2). |
| *Status* | Failed |
| *Test Data* | {"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from  KC,scbn\_3,DALLAS,46827,NEW,40,UNIT,1900-01-01KC,scbn\_3,FAIRFIELD,46827,NEW,20,UNIT,1900-01-01T00:00:00  KC,scbn\_3,LOSANGELES,46827,NEW,14,CASE,1900-01-01T00:00:00  KC,scbn\_3,SANJOSE,46827,NEW,14,UNIT,1900-01-01T00:00:00  KC,scbn\_3,SANFRANCISCO,46827,NEW,15,UNIT,2020-04-12T00:00:00  KC,scbn\_3,VISTA,46827,NEW,15,UNIT,2020-05-20T00:00:00  "} |
| *Expected Result* | {      "error": {          "statusCode": 400,          "message": "Mailformed CSV Input data found."      }  } |
| *Actual Result* | {      "error": {          "statusCode": 400,          "message": "The argument of the toMillis function must be an ISO 8601 formatted timestamp. Given \"1900-01-01KC\""      }  } |

# Malformed Data 2

|  |  |
| --- | --- |
| *Test Case* | Push malformed CSV Data inside the JSON Data to App Connect using API call. Changed the first row’s last column date format to yy-MM-dd where it was expected to be yyyy-MM-dd. |
| *Status* | Partially |
| *Test Data* | {"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from  KC,scbn\_3,DALLAS,46827,NEW,40,UNIT,19-01-01  KC,scbn\_3,FAIRFIELD,46827,NEW,20,UNIT,1900-01-01T00:00:00  KC,scbn\_3,LOSANGELES,46827,NEW,14,CASE,1900-01-01T00:00:00  KC,scbn\_3,SANJOSE,46827,NEW,14,UNIT,1900-01-01T00:00:00  KC,scbn\_3,SANFRANCISCO,46827,NEW,15,UNIT,2020-04-12T00:00:00  KC,scbn\_3,VISTA,46827,NEW,15,UNIT,2020-05-20T00:00:00  "} |
| *Expected Result* | {      "error": {          "statusCode": 400,          "message": "The ‘available from’ column value of Row (1) should be in yyyy-MM-dd format however the actual value is 19-01-01\""      }  } |
| *Actual Result* | {      "error": {          "statusCode": 400,          "message": "The argument of the toMillis function must be an ISO 8601 formatted timestamp. Given \"19-01-01\""  }  } |

# Mailslot ID Not Present in the Data

|  |  |
| --- | --- |
| *Test Case* | When sent the JSON data with MSlotID as “” then App Connect itself should fail the request. |
| *Status* | Partially |
| *Test Data* | {"MSlotID":"","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from  KC,scbn\_3,DALLAS,46827,NEW,40,UNIT,1900-01-01T00:00:00  KC,scbn\_3,FAIRFIELD,46827,NEW,20,UNIT,1900-01-01T00:00:00  KC,scbn\_3,LOSANGELES,46827,NEW,14,CASE,1900-01-01T00:00:00  KC,scbn\_3,SANJOSE,46827,NEW,14,UNIT,1900-01-01T00:00:00  KC,scbn\_3,SANFRANCISCO,46827,NEW,15,UNIT,2020-04-12T00:00:00  KC,scbn\_3,VISTA,46827,NEW,15,UNIT,2020-05-20T00:00:00  "} |
| *Expected Result* | {      "error": {          "statusCode": 400,          "message": "The mandatory field MSlotID is empty.”      }  } |
| *Actual Result* | {      "error": {          "statusCode": 400,          "message": "\"Supplier ID not found in Token service. Please onboard supplier\""      }  } |

# Buyer (Customer) ID Not Present in CSV Data

|  |  |
| --- | --- |
| *Test Case* | When the first row’s second column’s value is not present in the CSV data. |
| *Status* | Partially |
| *Test Data* | {"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from  KC,,DALLAS,46827,NEW,40,UNIT,1900-01-01T00:00:00  KC,scbn\_3,FAIRFIELD,46827,NEW,20,UNIT,1900-01-01T00:00:00  KC,scbn\_3,LOSANGELES,46827,NEW,14,CASE,1900-01-01T00:00:00  KC,scbn\_3,SANJOSE,46827,NEW,14,UNIT,1900-01-01T00:00:00  KC,scbn\_3,SANFRANCISCO,46827,NEW,15,UNIT,2020-04-12T00:00:00  KC,scbn\_3,VISTA,46827,NEW,15,UNIT,2020-05-20T00:00:00  "} |
| *Expected Result* | In absence of first row’s second column value, the App Connect should check that and fail the workflow.  {      "error": {          "statusCode": 400,          "message": "The first row\’s second column\’s value is mandatory to identify the IV buyer/customer, but it\’s not present.”      }  } |
| *Actual Result* | {      "error": {          "statusCode": 400,          "message": "\"NodeJs Service Failed with Error:\"   {\"message\":\"The SCBN buyer  could not be found. The buyer first needs to be onboarded in s4s-supplement-service.\"}"      }  } |

# When Token Generation NodeJS Services is Down

|  |  |
| --- | --- |
| *Test Case* | Give a wrong URL to simulate the NodeJS Service down scenario. Use the URL : https://s4s-supplement-service-dev1.mybluemix.net/s4s/iv-token |
| *Status* | **Passed** |
| *Test Data* | {"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from  KC,scbn\_3,FAIRFIELD,46827,NEW,20,UNIT,1900-01-01T00:00:00  KC,scbn\_3,LOSANGELES,46827,NEW,14,CASE,1900-01-01T00:00:00  KC,scbn\_3,SANJOSE,46827,NEW,14,UNIT,1900-01-01T00:00:00  KC,scbn\_3,SANFRANCISCO,46827,NEW,15,UNIT,2020-04-12T00:00:00  KC,scbn\_3,VISTA,46827,NEW,15,UNIT,2020-05-20T00:00:00  "} |
| *Expected Result* | {      "error": {          "statusCode": 404,          "message": " Requested route ('s4s-supplement-service-dev1.mybluemix.net') does not exist.”      }  } |
| *Actual Result* | {      "error": {          "statusCode": 400,          "message": Requested route ('s4s-supplement-service-dev1.mybluemix.net') does not exist }"      }  } |

# Customer ID is Not Onboarded in MongoDB

|  |  |
| --- | --- |
| *Test Case* | Customer Id is present in CSV file but not onboarded in Mongo DB |
| *Status* | **Passed** |
| *Test Data* | {"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from  KC,scbn\_10,DALLAS,46827,NEW,40,UNIT,1900-01-01T00:00:00  KC,scbn\_3,FAIRFIELD,46827,NEW,20,UNIT,1900-01-01T00:00:00  KC,scbn\_3,LOSANGELES,46827,NEW,14,CASE,1900-01-01T00:00:00  KC,scbn\_3,SANJOSE,46827,NEW,14,UNIT,1900-01-01T00:00:00  KC,scbn\_3,SANFRANCISCO,46827,NEW,15,UNIT,2020-04-12T00:00:00  KC,scbn\_3,VISTA,46827,NEW,15,UNIT,2020-05-20T00:00:00  "} |
| *Expected Result* | {      "error": {          "statusCode": 400,          "message": "\"NodeJs Service Failed with Error:\"   {\"message\":\"The SCBN buyer scbn\_10 could not be found. The buyer first needs to be onboarded in s4s-supplement-service.\"}"  }  } |
| *Actual Result* | {      "error": {          "statusCode": 400,          "message": "\"NodeJs Service Failed with Error:\"   {\"message\":\"The SCBN buyer scbn\_10 could not be found. The buyer first needs to be onboarded in s4s-supplement-service.\"}"      }  } |

# Supplier ID returned by Token Service is not matching with the Supplier ID present in payload

|  |  |
| --- | --- |
| *Test Case* | Customer Id is present in CSV file but not onboarded in Mongo DB |
| *Status* | **Passed** |
| *Test Data* | {"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from  KC1,scbn\_3,DALLAS,46827,NEW,40,UNIT,1900-01-01T00:00:00  "} |
| *Expected Result* | {      "error": {          "statusCode": 400,          "message": "\"Supplier ID from token service. is not matching with Supplier ID in Payload"      }  } |
| *Actual Result* | {      "error": {          "statusCode": 400,          "message": "\"Supplier ID from token service. is not matching with Supplier ID in Payload"      }  } |

# Provide malformed date in any of the date column of any of the row in CSV payload

## Test Case:

Give wrong date for ‘available from’ column.

## Status: Partially

## Test Data Used:

{"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from

KC1,scbn\_3,DALLAS,46827,NEW,40,UNIT,1900-01-

"}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\"Invalid Date in ‘available from’ column"

    }

}

## Actual Result:

{

    "error": {

        "statusCode": 400,

        "message": "The argument of the toMillis function must be an ISO 8601 formatted timestamp. Given \"1900-01-0\""

    }

}

# Validate ‘type’ field inserted in IV.

## Test Case:

Validate data in IV either manually or by calling IV API to check that App Connect is inserting the data correctly for following conditions "available from" Is greater than current date then set type = WIP else ONHAND" "available from" is less than current date then set eta = "1900-01-01T00:00:00Z" else "available from"

## Status: Passed

## Test Data Used:

{"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from

KC,scbn\_3,DALLAS,46827,NEW,40,UNIT,1900-01-01

KC,scbn\_3,VISTA,46827,NEW,40,UNIT,2021-01-01

"}

## Expected Result:

For Item ID -DALLAS::46827

[

    {

        "itemId": "KC::46827",

        "unitOfMeasure": "UNIT",

        "productClass": "NEW",

        "shipNode": "KC::DALLAS",

        "type": "ONHAND",

        "quantity": 30.0,

        "eta": "1900-01-01T00:00:00.000Z",

        "shipByDate": "2500-01-01T00:00:00.000Z"

    }

]

For Item ID -VISTA::46827

[

    {

        "itemId": "KC::46827",

        "unitOfMeasure": "UNIT",

        "productClass": "NEW",

        "shipNode": "KC::VISTA",

        "type": "WIP",

        "quantity": 15.0,

        "eta": "2020-05-20T00:00:00.000Z",

        "shipByDate": "2500-01-01T00:00:00.000Z"

    }

]

## Actual Result:

For Item ID -DALLAS::46827

[

    {

        "itemId": "KC::46827",

        "unitOfMeasure": "UNIT",

        "productClass": "NEW",

        "shipNode": "KC::DALLAS",

        "type": "ONHAND",

        "quantity": 30.0,

        "eta": "1900-01-01T00:00:00.000Z",

        "shipByDate": "2500-01-01T00:00:00.000Z"

    }

]

For Item ID -VISTA::46827

[

    {

        "itemId": "KC::46827",

        "unitOfMeasure": "UNIT",

        "productClass": "NEW",

        "shipNode": "KC::VISTA",

        "type": "WIP",

        "quantity": 15.0,

        "eta": "2020-05-20T00:00:00.000Z",

        "shipByDate": "2500-01-01T00:00:00.000Z"

    }

]

What if:

When the invalid / incomplete rows are present in CSV payload and missing mandatory data fields like:

1. Item ID

## a. Test Case:

Empty value for ‘Item ID’ column.

## Status: Failed

## Test Data Used:

{"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from

KC1,scbn\_3,DALLAS,,NEW,40,UNIT,1900-01-

"}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\"Item id is missing"

    }

}

## Actual Result:

{

    "MSlotID": "SJNG7A01",

    "Payload": "\"Sync Supply  called Successfully\""

}

**Note:- No Validation at IV side if Item ID is not passed.**

1. Quantity

## a. Test Case:

Empty value for ‘Quanity’ column.

## Status: Failed

## Test Data Used:

{"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from

KC1,scbn\_3,DALLAS,36428,NEW, ,UNIT,1900-01-

"}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\" Quantity is missing"

    }

}

## Actual Result:

{

    "MSlotID": "SJNG7A01",

    "Payload": "\"Sync Supply  called Successfully\""

}

**Note:-Get supplies API is not returning that particular Itemid, ship node .So, that particular row is not getting updated/ updated with zero quantity.**

1. Location ID

## a. Test Case:

Empty value for ‘location id column.

## Status: Partially

## Test Data Used:

{"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from

KC1,scbn\_3, ,36428,NEW,20,UNIT,1900-01-

"}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\" Location id is missing"

    }

}

## Actual Result:

{

    "error": {

        "statusCode": 400,

        "message": "\"IV Sync Supply Failed With Error\":  IV URL-https://api.watsoncommerce.ibm.com/inventory/eu-87d7c9b0/v1/supplies : {\"error\_message\":\"Parameter: [shipNode] has invalid value: [KC::] due to cause: [Provided Ship Node does not exist]\"} : "

    }

}

1. Available From

## a. Test Case:

Empty value for ‘available from’ column.

## Status: Partially

## Test Data Used:

{"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from

KC1,scbn\_3,DALLAS ,36428,NEW,20,UNIT, -

"}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\" Available from is missing"

    }

}

## Actual Result:

{

    "error": {

        "statusCode": 400,

        "message": "The argument of the toMillis function must be an ISO 8601 formatted timestamp. Given \"\""

    }

}

1. Supplier ID

## a. Test Case:

Empty value for ‘supplier id’ column.

## Status: Partially

## Test Data Used:

{"MSlotID":"SJNG7A01","Payload" :"supplier id, customer id, location id, item id,product class, quantity,uom,available from

,scbn\_3,DALLAS ,36428,NEW,20,UNIT,1900-01-01

"}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\" Supplier ID is missing"

    }

}

## Actual Result:

{

    "error": {

        "statusCode": 400,

        "message": "\"Supplier ID from token service. is not matching with Supplier ID in Payload"

    }

}

1. No IV URL

## a. Test Case:

Empty URL value from Token service.

## Status: Partially

## Test Data Used:

{

    "url": "",

    "buyer\_scbn\_id": "scbn\_3",

    "supplier\_id": "KC",

    "iv\_tenant\_id": "eu-87d7c9b0",

    "bearer\_token": "7Yp104qgC8rIcXFLZsoWvVAaPzmxwQHu",

    "time\_to\_live": 43170

}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\" No IV url returned by IV service "

    }

}

## Actual Result:

{

    "error": {

        "statusCode": 400,

        "message": "The URL you provided is invalid: \"\"\\\" \\\"\"\".",

        "errorDetail": "{\"message\":\"Invalid URL: \\\" \\\"\",\"input\":\"\\\" \\\"\"}"

    }

}

1. No Bearer Token

## a. Test Case:

Empty bearer token value from Token service.

## Status: Partially

## Test Data Used:

{

     "url": " https://api.watsoncommerce.ibm.com/inventory/eu-8d7c9b0/v1/supplies",

    "buyer\_scbn\_id": "scbn\_3",

    "supplier\_id": "KC",

    "iv\_tenant\_id": "eu-87d7c9b0",

    "bearer\_token": "",

    "time\_to\_live": 43170

}

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\" No Bearer returned by IV service "

    }

}

## Actual Result:

{

    "error": {

        "statusCode": 400,

         "message": "You cannot consume this service"

    }

}

1. IV is down or a wrong / malformed IV URL supplied by Token Service.

## a. Test Case:

URL supplied by Token service is wrong.

## Status: Passed

## Test Data Used:

## Expected Result:

{

    "error": {

        "statusCode": 400,

        "message": "\"IV Sync Supply Failed With Error\":  IV URL-https://api.watsoncommerce.ibm.com/inventory/eu-87d7c9b0/v1/supplies1 :  : "

    }

}

## Actual Result:

{

    "error": {

        "statusCode": 400,

        "message": "\"IV Sync Supply Failed With Error\":  IV URL-https://api.watsoncommerce.ibm.com/inventory/eu-87d7c9b0/v1/supplies1 :  : "

    }