EXPANDABLE ERP INTEGRATION

EXPANDABLE RESTFUL WEB SERVICES APIS DESIGNER TIPS AND TROUBLESHOOTING

Expandable Software, Inc.
900 Lafayette Street 4th Floor
Santa Clara, CA 95050
(408) 261-7880
www.expandable.com

Table of Contents

Related Documentation

1. Recorded Webinar Overview: Watch this ~60 min overview for high-level overview and demo of the Web Services

   http://www.ieuga.org/downloads/webinars/2014-10-01_10.01_Expandable_Announces_RESTful_Web_Services.wmv

2. On-Line Web Services: ExpandableWebAPI/Help on the Evaluation site
   a. Visit this Customer Evaluation Site URL: http://www.expandable.com/demos/demo-room/web-services.html to view on-line installed product help

3. "***WebService JSON DatastructureOrder***", Excel Workbook

4. Expandable RESTful Web Services Data Sheet

5. Expandable RESTful Web Service Overview Slides, ask your Expandable Sales Manager to provide a copy of the overview.

6. Expandable "**table-list-and-layouts V9d2d1**" System Documentation

7. Expandable ERP/MRP "HELP" – Main ERP Client/Server Application

8. Expandable User Documentation and Training Manuals
   a. Visit Expandable's "Customer Resource Center" for User manuals documentation and support notes: http://www.expandable.com/support-login.asp

Change History

Date	Version	Change Editor	Description
11/12/2015	1.0	D Payton	Capture of some common issues and resolutions
12/29/2015	1.1	D Payton	Add reference to "***WebService JSON DatastructureOrder***", Excel Workbook

Creating and Using a Session Tokin

The key here is the Expandable Session Authorization Tokin “AuthTokin” reside in the HTTP Response Header or needs to be placed in the HTTP Post Header for requests sent to the RESTful Web Services. The Body of the HTTP Response/Post will contain the working JSON **"Name": "Value"** pairs for the transaction data.

JSON Structural Error Messages

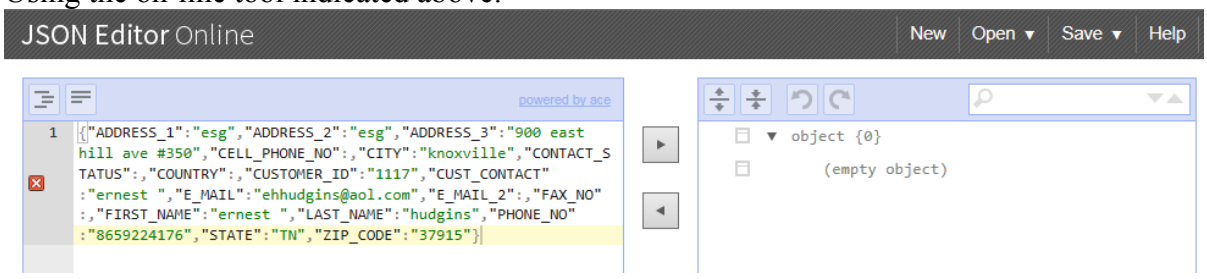
The dreaded: "`Error: 400/Request JSON is invalid.`" This error message really has nothing to do with the Business Logic of the Web Service API and is generally a direct miss-coding of the design to adhere to standard JSON structures and syntax. The best way to evaluate this is to use a standard JSON parser. A decent On-Line tool like: http://www.jsoneditoronline.org/ (one of many out there) which makes it much easier to see issues and parsers will point out JSON structural & syntactical errors as well.

A Designer's Example 1:

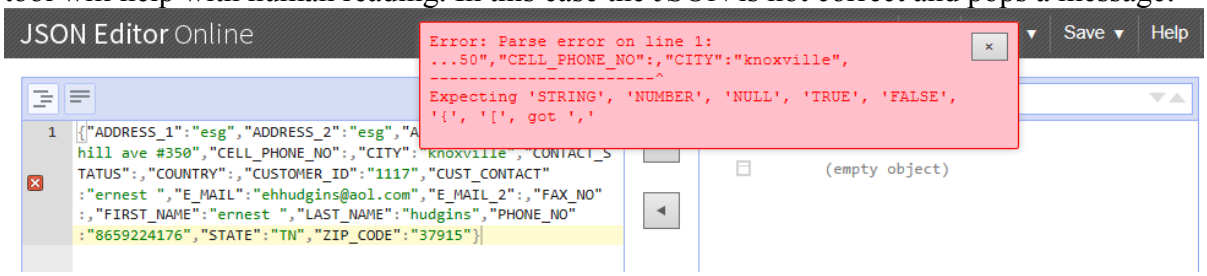
```
DataPost:
{"ADDRESS_1":"esg","ADDRESS_2":"esg","ADDRESS_3":"900 east hill ave
#350","CELL_PHONE_NO":,"CITY":"knoxville","CONTACT_STATUS":,"COUNTRY":,"CUSTOMER_ID":"1117","
CUST_CONTACT":"ernest
","E_MAIL":"ehhudgins@aol.com","E_MAIL_2":,"FAX_NO":,"FIRST_NAME":"ernest
","LAST_NAME":"hudgins","PHONE_NO":"8659224176","STATE":"TN","ZIP_CODE":"37915"}

Error: 400/Request JSON is invalid.
```

Using the on-line tool indicated above:



This tool points out an error right away  indicating where an issue with JSON resides, still we can attempt to make this a bit more readable by clicking  and if the JSON is right the tool will help with human reading. In this case the JSON is not correct and pops a message:



Indicating the issue,

```
Error: Parse error on line 1:
...50","CELL_PHONE_NO":,"CITY":"knoxville",
----------------------^
Expecting 'STRING', 'NUMBER', 'NULL', 'TRUE', 'FALSE', '{', '[', got ','
```

And what is expected for JSON syntax to be correct and the designer can correct code to deliver the appropriate JSON. As a note this designer has a number of issues where the **"Name": "Value"** pairs are missing "Value" and the tool errors at the first one.

Business Logic Triggers – Data Order Dependency

Most of the Expandable base business logic can be Data Order Dependent, for Expandable default recommended data order see: "***WebService JSON DatastructureOrder***", Excel Workbook

As an example, this JSON works fine in the order delivered by the this JSON structure for Inventory Issues Entry/Reversal (ICTIE), in standard Expandable ERP GUI for Inventory Issues Entry/Reversal (ICTIE) the system doesn't allow the JOB information until the Part information is entered:

```
{
 "ACTION_TYPE": "IR",
 "STORES_CODE": "FG",
 "PART_ID": "12-60036",
 "LOT_ID":"SP-10002",
 "JOB_ID": "500023",
 "OPER_CODE":"10",
 "QUANTITY": "5"
}
```

But this JSON data order will generate an ERROR, with the same data, just with the order of data changed with JOB related data coming before the PART data:

```
{
 "ACTION_TYPE": "IR",
 "JOB_ID": "500023",
 "OPER_CODE":"10",
 "PART_ID": "12-60036",
 "LOT_ID":"SP-10002",
 "QUANTITY": "5",
 "STORES_CODE": "FG"
}
```

In this case the servce returns an error as follows:

Status

500/Must enter a value for the required field GL Account (GLACCOUNT).

Headers

Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/vnd.expandable.api.v930+json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/7.5
AuthToken: 8cb04d31-931c-4909-81ab-e58cd2305ffe
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 11 Nov 2015 00:00:25 GMT

Content-Length: 272

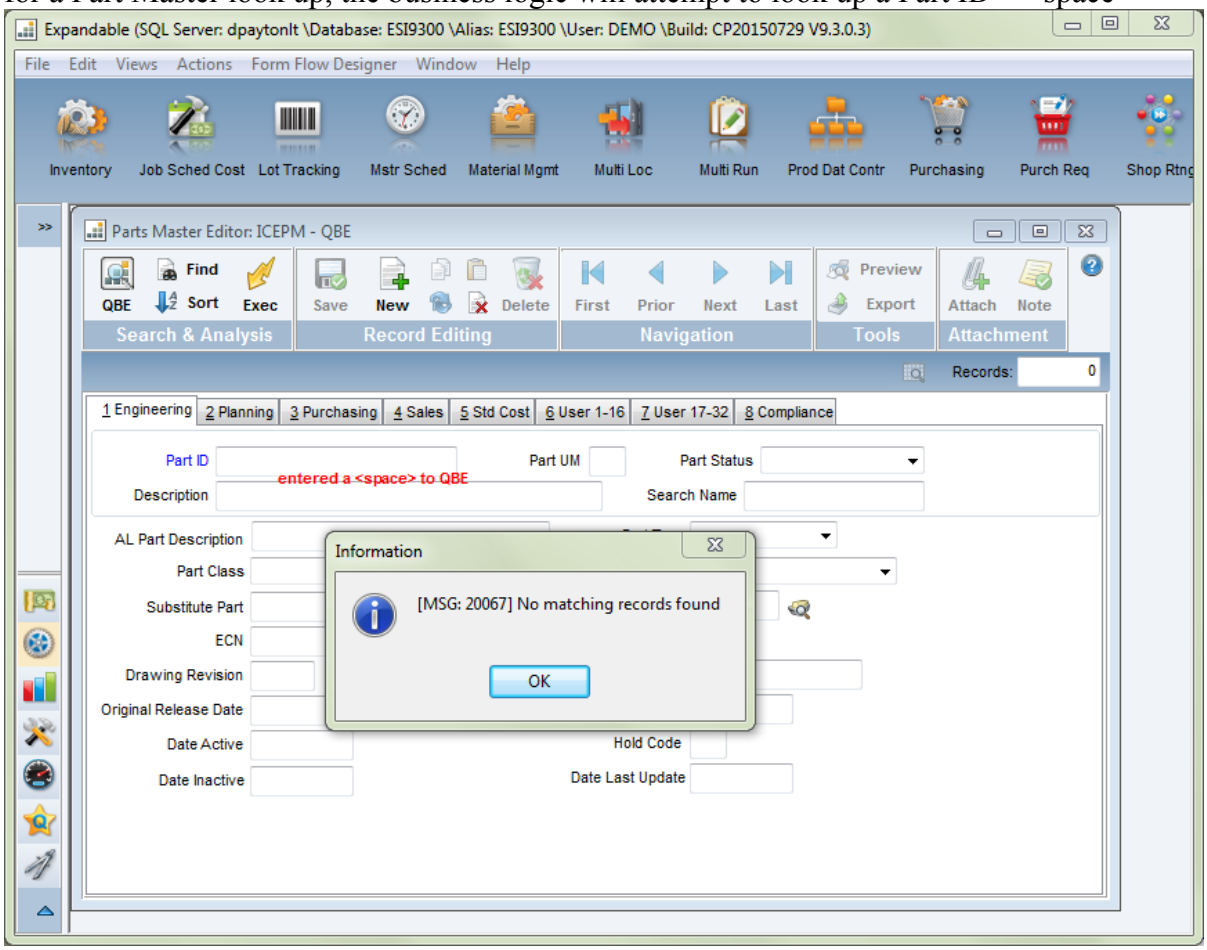
Body

{"Message":"Must enter a value for the required field GL Account (GLACCOUNT).","MessageCode":"DATA_ELEMENT_VALUE_FAILS_REQUIRED_CHECK","MessageData":"ACTION_TYPE='IR', JOB_ID='500023', OPER_CODE='10', PART_ID='12-60036', LOT_ID='SP-10002', QUANTITY='5', STORES_CODE='FG'"}

“No Data” Blank verse White Space Values

A no-data value is represented by two single straight quotes for JSON null data set if the desired effect to deliver no-data other ASCII white space characters can look like no data but are in fact data and treated as such for look ups.

This is the same as if an Expandable GUI user attempted to enter a <space> in the GUI editor for a Part Master look up, the business logic will attempt to look up a Part ID = <space>



A designer sent JSON that was causing errors:

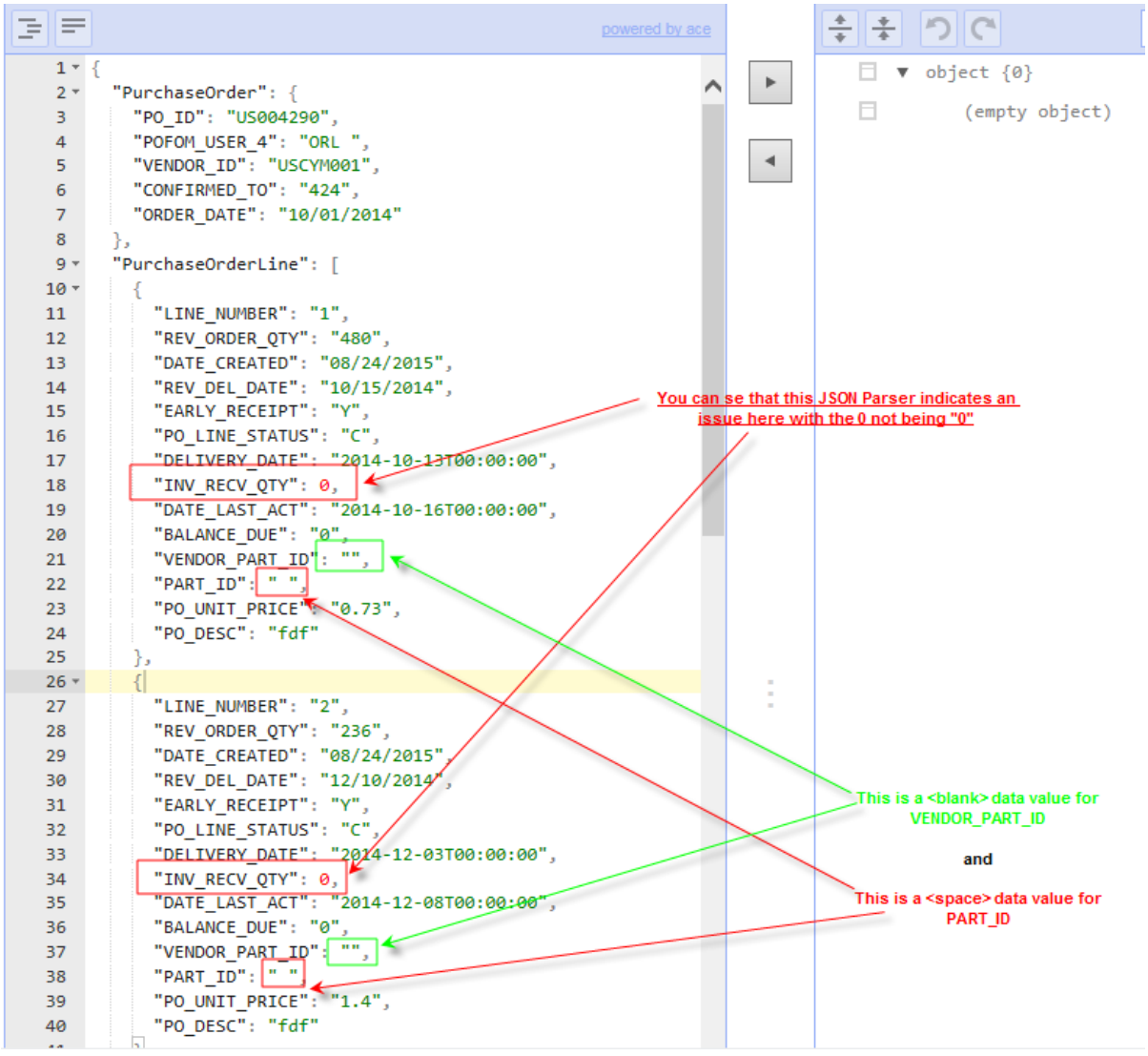
```
{"PurchaseOrder":{"PO_ID":"US004290","POFOM_USER_4":"ORL
","VENDOR_ID":"USCYM001","CONFIRMED_TO":"424","ORDER_DATE":"10/01/2014"},"PurchaseOrderLine":
[{"LINE_NUMBER":"1","REV_ORDER_QTY":"480","DATE_CREATED":"08/24/2015","REV_DEL_DATE":"10/15/2
014","EARLY_RECEIPT":"Y","PO_LINE_STATUS":"C","DELIVERY_DATE":"2014-10-
13T00:00:00","INV_RECV_QTY":0,"DATE_LAST_ACT":"2014-10-
16T00:00:00","BALANCE_DUE":"0","VENDOR_PART_ID":"","PART_ID":"
","PO_UNIT_PRICE":"0.73","PO_DESC":"fdf"},{"LINE_NUMBER":"2","REV_ORDER_QTY":"236","DATE_CREA
TED":"08/24/2015","REV_DEL_DATE":"12/10/2014","EARLY_RECEIPT":"Y","PO_LINE_STATUS":"C","DELIV
ERY_DATE":"2014-12-03T00:00:00","INV_RECV_QTY":0,"DATE_LAST_ACT":"2014-12-
08T00:00:00","BALANCE_DUE":"0","VENDOR_PART_ID":"","PART_ID":"
","PO_UNIT_PRICE":"1.4","PO_DESC":"fdf"},{"LINE_NUMBER":"3","REV_ORDER_QTY":"304","DATE_CREAT
ED":"08/24/2015","REV_DEL_DATE":"12/10/2014","EARLY_RECEIPT":"Y","PO_LINE_STATUS":"C","DELIVE
RY_DATE":"2014-12-03T00:00:00","INV_RECV_QTY":0,"DATE_LAST_ACT":"2014-12-
08T00:00:00","BALANCE_DUE":0}],"PurchaseOrderComment":[{"PO_LINE_NO":"1","PRINT_PO":"NP","COM
MENT_TEXT":"aaa"},{"PO_LINE_NO":"2","PRINT_PO":"NP","COMMENT_TEXT":"aaa"}],"PurchaseOrderExte
ndedUser":[{"LINE_NUMBER":"1","POFDXU_USER_10":"3994","POFDXU_USER_11":"1","POFDXU_USER_14":"
```

```
","POFDXU_USER_43":" ","POFDXU_USER_29":" ","POFDXU_USER_1":"BOS","POFDXU_USER_45":"
","POFDXU_USER_13":"N","POFDXU_USER_42":"","POFDXU_USER_2":"WAF ","POFDXU_USER_44":"
","POFDXU_USER_17":"
"},{"LINE_NUMBER":"2","POFDXU_USER_10":"3995","POFDXU_USER_11":"1","POFDXU_USER_14":"
","POFDXU_USER_43":" ","POFDXU_USER_29":" ","POFDXU_USER_1":"BOS","POFDXU_USER_45":"
","POFDXU_USER_13":"N","POFDXU_USER_42":"","POFDXU_USER_2":"WAF ","POFDXU_USER_44":"
","POFDXU_USER_17":"
"},{"LINE_NUMBER":"3","POFDXU_USER_10":"3995","POFDXU_USER_11":"2","POFDXU_USER_14":"
","POFDXU_USER_43":" ","POFDXU_USER_29":" "}]}
```


First of all, this is readable but much easier if copy/pasted to a JSON parser. Here is a screen shot of JSON issue from the On-Line tool: http://www.jsoneditoronline.org/ which makes it much easier to see issues and many parsers will point out JSON structural & syntactical errors as well. Even here is can be hard to tell between:

"PART_ID": " " and "PART_ID": ""

But it is a bit easier…



1. The PART_ID is not being sent as blank (No Data) but rather as a <space> had to see but there is a space in the "Name":"Value" pair take a look at the VENDOR_PART_ID –vs- PART_ID:

   "VENDOR_PART_ID": "",
   "PART_ID": " ",
   "PO_UNIT_PRICE": "0.73",
   "PO_DESC": "fdf"

   This one is not so subtle in JSON as it is formed correctly but we are looking for a part with the value of a single <space>, we probably should trim spaces but we don't do that in ERP/MRP either (if I do a QBE on a PART_ID of space I get no matching records message). This will correct the issue by delivering no data <blank> in the PART_ID data value:

   "VENDOR_PART_ID": "",
   "PART_ID": "",
   "PO_UNIT_PRICE": "0.73",
   "PO_DESC": "fdf"

In the example designer's JSON above, there are a number of cases where the <space> was used to represent "no-data" in some of the user fields etc. these are may have less impact for the particular Expandable System has these fields defended as optional; however, if they are defined in the dictionary as Mandatory Drop Down Selection the business logic may also not find a <space> in the dictionary look up table and reject the transaction.

2. There is also a JSON issue with "INV_RECV_QTY": 0 …. The 0 should be "0" to delimit the value appropriately for JSON structures. I would have assumed you'd get a '400' JSON error from that before even getting to the business logic but maybe the JSON read is a little bit forgiving generally it would cause a malformed JSON structure error.

Web Service and Business Logic - Date Formats Local Server's "Region Control Panel"

WS/BLS handling of Date Formats should align with the server that the Web Services is installed on (IIS Server). Just an important note through, the error message generated when submitted date format doesn't match, can be misleading in that the "MM/DD/YYYY" is hard coded in the error message and not dynamically generated from the local regional setting for the date setting on the IIS server. So a designer needs to look at the "Region Control Panel" and see what the date format is set and also review code for the JSON date data being submitted to make sure the formats are consistent. So in summary:

1. WS/BLS will adhere to the local regional date format set in the "Region Control Panel" of the IIS server when processing input data (JSON Date Values).

2. If the format is mismatched the error message would be potentially misleading as we hard coded the MM/DD/YYYY as the format in the error message.

Sales Order Creation DATA_ELEMENT_VALUE_FAILS_ALPHANUMERIC_CHARACTERISTIC_CHECK

Some busines logic messages maybe generated that potentially don't make sense on first blush, but Web Services mimics the standard Editors and for example when a CUSTOMER_ID is entered pulls in to the editor (generally as informational, read only) the CUSTOMER_NAME. In the case of a Sales Order, for exmaple, the CUSTOMER_ID representing the SOLD_TO for the order and the SHIP_TO customer ID pull in the coresponding Customer Name and the Web Service inforces the dictionary limits and for example if a message like this is generated from the SalesOrder/Create (or Update):

```
{
    "Message": "The value for Customer Name must be no more
than 30 characters long.",
    "MessageCode":
"DATA_ELEMENT_VALUE_FAILS_ALPHANUMERIC_CHARACTERISTIC_CHECK",
    "MessageData": "SO_TYPE='NS', SOFOM_USER_10='BLUM, KARL',
SOFOM_USER_3='ACL', SOFOM_USER_1='AOI1', ORDER_TAKER='MWF',
SALESMAN_ID='AOI', SOFOM_USER_11='W1A9C1516A3A18EE',
CUST_PO_ID='mwftestpo25', SHIP_TO_CUST='BLO002',
SHIP_NOTE_2='Echo de0023', BILL_TO_CUST='BLO002',
SHIP_NOTE_1='12/03/2015', CUSTOMER_ID='BLO002',
ORDER_CLASS='N'"
}
```

Make sure that all the Sales Order Customer Names in the dictionary are set to appropreatly defined lengths as follows should clear this error:

```
UPDATE XXFDIC SET DESCRIPTION='60' WHERE DATA_ELEM_NAME='CUSTOMER_NAME' AND
ENTRY_TYPE='C'
UPDATE XXFDIC SET DESCRIPTION='60' WHERE DATA_ELEM_NAME='BILL_NAME' AND
ENTRY_TYPE='C'
UPDATE XXFDIC SET DESCRIPTION='60' WHERE DATA_ELEM_NAME='SHIP_NAME' AND
ENTRY_TYPE='C'
```

Appendix: Common Information & Error Messages

Error/Status Message	Explanation and Corrective Action
4XX/*	The 400's messages are generally issues or message encountered before the Business Logic layer. For example: - missing log in credentials, - session AuthToken, - invalid JSON data structures that can't be parsed because of missing/incorrect syntax, - etc.
400/Request JSON is invalid.	The submitted JSON data structure is invalid syntax or general structure. Check the programming for proper JSON formation of the JSON **name:value** pairs and syntax structuring.
401/AuthToken was not provided.	Web Services must have a valid session Token generated by a "api/Security/Login (POST)" with valid security credentials. Each Service call requires the Token to be provided in the call's HEADER as the "AuthToken"
403/AuthToken is not known.	Web Services must have a valid session Token generated by an "api/Security/Login (POST)" with valid security credentials. Each Service call requires the Token to be provided in the call's HEADER as the "AuthToken". This return message indicates that an "AuthToken" was provided but is not registered as an active or valid Token. The Token may have expired in which case the calling program would need to resubmit login credentials via "api/Security/Login (POST)" to obtain a new session Token.
5XX/*	The 500's levels of messages generally are from the Web Service core business logic and should provide enough information in the Body of the return JSON to indicate the issue. Message may also indicate Licensing or System User Rights issue for a given service call. For example if the Customer has not licensed a particular feature or the system user rights have not been setup or the module SYS_INSTALL record has not been created to enable the module in the system. These would generate 5XX type errors
500/Program *<program_code>* **is not allowed as feature.** **Example:** 500/Program SOEOE is not allowed as feature.	BLS Feature Key likely missing or it doesn't contain the Lic Key for enabling the <program_code> Resolve by installing the "BLS Feature Key" in to the Expandable Web Service IIS instance. If the customers BLS Feature Key is installed contact Expandable for Licensing of feature that has been blocked.

500/A user must be logged in before this action can be performed.	
500/ <program_name (program_code)> **search criteria** <criterial_value> **does not match any records.** **Example:** 500/Customer Master Editor (SOECM) search criteria CUSTOMER_ID='ABCCORP' does not match any records.	The search criteria provided to the program didn't match any data in the target program tables and therefore didn't return any results.

User Notes: