



Enable™ Version 10

Custom Widget Creation Guide

Revised 8/2/2019

Enable 10.0 Release Enable Suite - All components

- **EnterworksInstallPackage_20190423_1624**
- **EnterworksDB_20190423_1624**
- **EnterworksDBScripts_20190423_1624**

Enable Patch

- **EnableServer_10.0_Patch_20190423_1549**

EnterWorks®, Acquisition Inc. a Winshuttle Company

46040 Center Oak Plaza Suite 115

Sterling, VA 20166

©EnterWorks Acquisition, Inc.
Loudoun Tech Center
46040 Center Oak Plaza
Suite 115
Sterling, VA 20166

1.888.242.8356 (Sales and General Information)
1.888.225.2705 (U.S. Support)
<http://www.enterworks.com>

EnterWorks® *Enable 10 Custom Widget Creation Guide*

Copyright © 2007-2019 EnterWorks Acquisition, Inc., a Winshuttle company. All rights reserved.

Law prohibits unauthorized copying of all or any part of this document. Use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in FAR 52.227-14.

“EnterWorks” and the “EnterWorks” logo are registered trademarks and “Enable PIM”, “EnterWorks Process Exchange” and “EnterWorks Product Information Management” are trademarks of EnterWorks Acquisition, Inc.

Windows, .Net, IIS, SQL Server, Word, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Java and all Sun and Java based trademarks are trademarks or registered trademarks of the Oracle Corporation in the United States and other countries.

Oracle is a registered trademark and Oracle 10g is a trademark of Oracle Corporation.
Pentium is a registered trademark of Intel Corporation in the United States and other countries.
JBoss is a registered trademark of Red Hat, Inc.
All other trademarks and registered trademarks are the property of their respective holders.

All icons and graphics, with the exception of the "e." logo, were obtained from West Coast Icons and Design at <http://www.bywestcoast.com>. EnterWorks Acquisition, Inc. retains copyrights for all graphics unless otherwise stated. All other trademarks and registered trademarks are the property of their respective holders.

This document is furnished for informational purposes only. The material presented in this document is believed to be accurate at the time of printing. However, EnterWorks Acquisition, Inc. assumes no liability in connection with this document except as set forth in the License Agreement under which this document is furnished.

Table of Contents

1	Document Conventions	4
2	Customer Support	5
3	Overview	6
4	Example Application	7
5	Getting Started.....	7
5.1	Set Up the HTML Page.....	7
5.2	Point to the Custom Action JSP/HTML.....	8
5.3	Check the Attribute Set Up	10
6	Write the HTML/JSP for the Enable 2020 UI	11
6.1	Pull Data	11
6.2	Rest API Calls	11
6.3	Classic UI HTML/JSP	12
7	Appendix	13
7.1	itemDataMap	13

1 Document Conventions

This EnterWorks document uses the following typographic conventions:

Convention	Usage
pathnames	Pathnames are shown with backslashes, as for Windows systems.
Courier New font	<p>Denotes sample code, for example, Java, IDL, and command line information. May be used to denote filenames and pathnames, calculations, code samples, registry keys, path and file names, URLs, messages displayed on the screen.</p> <p>If <i>italicized</i> and in angle brackets (< >), it denotes a variable.</p>
Calibri Font (bold)	<p>When used in body text, it denotes an object, area, list item, button, or menu option within the graphical user interface; or a database name or database-related object. (Examples: the Save button; the Product tab; the Name field; the SKU repository)</p> <p>Can also be used to denote text that is typed in a text box. (Example: Type “trackingNo” in the Name field)</p>
Blue underlined text	Words, phrases or numbers in blue are active links that can be clicked. Clicking these active links will bring the user to the required information, steps, pages chapters, or URL.

2 Customer Support

There's a reason why EnterWorks receives the highest customer satisfaction scores from firms including Gartner® and Forrester® – our customers are our top priority.

EnterWorks provides a full spectrum of customer support. Check your maintenance contract for the level of support purchased. The first time you contact customer support, you'll receive a customer identification number. Keep this number for future reference when using the EnterWorks customer support service.

How to reach us	Comments
<p>On the Web:</p> <p>https://enterworkssupport.zendesk.com</p>	<p>For detailed discussions of hardware, software, configuration issues, or Helpdesk credentials, contact your EnterWorks representative.</p>

3 Overview

This document covers the design and implementation of custom actions in the Enable Product Information Management (PIM) system. The Enable system is configured for each organization's product requirements so not all functionality is applicable for every organization. If a function described in this document is desired but not available, contact the Systems Administrator.

For administrative functionality, see the *Enable 10 Administrator Guide*.

For user functionality, see the *Enable 10 User Guide*.

For API functionality, see the documentation available in the Enable 2020 User Interface.

4 Example Application

Custom actions are usually written when a client needs to perform an action on an attribute. Here is an example:

Objective: CandyCo needs a way to create a new product with the same properties as its parent item but with one difference: the name of the item will be created as [analog/flavor] + [parent name]

Example: Midnight Fudge (parent) > Cherry Midnight Fudge or Peanut Butter Midnight Fudge.

Attribute: The analogs are saved on a list that belongs to an item. The analogs will describe different versions of that item.

Custom Action: When a user selects the custom action button a dialog will appear that contains a list of available analogs (flavors). The user selects the flavors they want to use to create an item, and then they click the **Create** button. The program creates a new flavor (assuming one doesn't already exist) for each flavor selected.

Action Complete.

5 Getting Started

In this section we will configure a custom action button both in the Enable UI as well as in the Enable 2020 UI. The steps to do so are:

1. Set up an HTML page to gather user input.
2. Set up a custom action option on the relevant attribute.
3. Test our solution.

5.1 Set Up the HTML Page

The first step in configuring a custom action button is to set up an HTML page to collect user input.

1. Set up an HTML page in the following folder:

```
\\<server>\d$\Enterworks\EnableServer\tomcat\webapps\w  
ebcm\custom
```

2. Create a JSP file in the same folder and name it accordingly. For example:
AnalogAction.jsp.

Name	Date modified	Type	Size
jquery	4/2/2019 2:10 PM	File folder	
ScheduledActivities	4/30/2019 4:00 AM	File folder	
AnalogAction.jsp	4/4/2019 4:01 PM	JSP File	11 KB
AnalogActionService.jsp	4/9/2019 3:16 PM	JSP File	8 KB
AttrCalculateNoPopupExample1.jsp	4/30/2019 4:00 AM	JSP File	5 KB
AttrCalculateNoUserInterfaceExample1.jsp	4/30/2019 4:00 AM	JSP File	5 KB
AttrCalculatePopupExample1.jsp	4/30/2019 4:00 AM	JSP File	9 KB
AttributeHTMLEditor.jsp	4/30/2019 4:00 AM	JSP File	9 KB
AutoClassificationExample.jsp	4/30/2019 4:00 AM	JSP File	16 KB
CallValidationForLinkedItems.jsp	4/30/2019 4:00 AM	JSP File	27 KB

Don't worry too much about what is in the JSP file for now; we just want it there so that we can point to it.

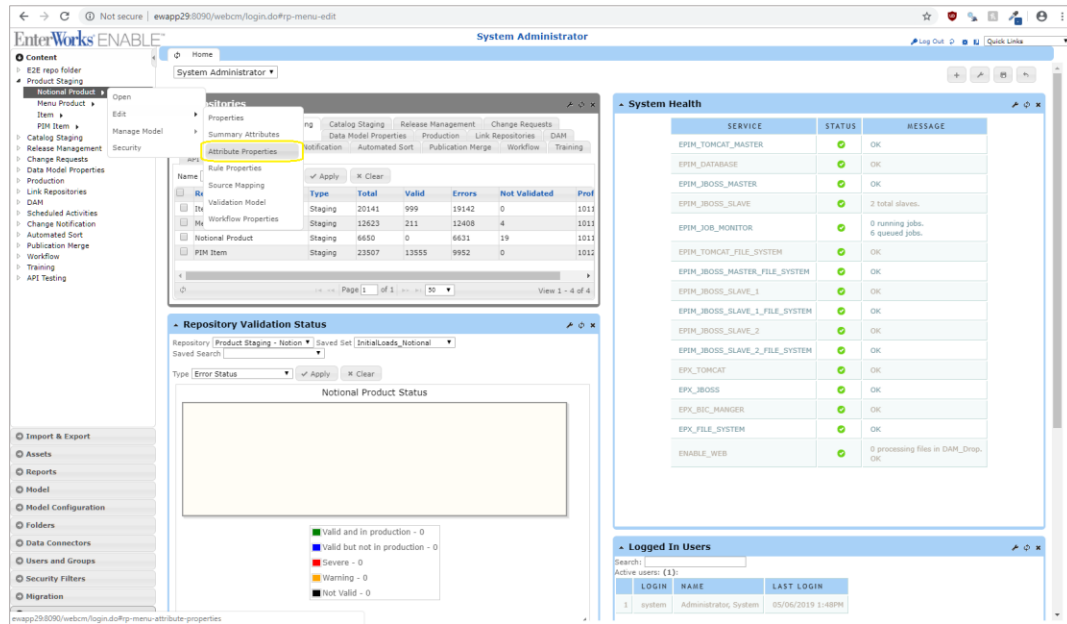
3. Create a copy of the JSP file and change its extension to an HTML extension. This must be done because the Enable 2020 UI looks for an HTML file rather than JSP file.
4. Move the new HTML file to the following folder where the Enable 2020 UI will access it:
`\\<server>\d$\Enterworks\enable2020\custom.`

Name	Date modified	Type	Size
bak	5/6/2019 1:56 PM	File folder	
jq	4/12/2019 1:07 PM	File folder	
jquery	4/2/2019 4:05 PM	File folder	
AnalogAction.html	4/16/2019 11:11 AM	Chrome HTML Document	14 KB
GoogleApiTest.html	8/17/2018 9:56 AM	Chrome HTML Document	4 KB
jquery-ui.min.css	4/2/2019 2:03 PM	Cascading Style Sheet Document	30 KB
MarketExplosion.html	4/19/2019 11:57 AM	Chrome HTML Document	32 KB
MenuProductItemExplosion.html	4/10/2019 1:28 PM	Chrome HTML Document	18 KB
MenuProductReleaseExplosion.html	4/10/2019 11:52 AM	Chrome HTML Document	15 KB

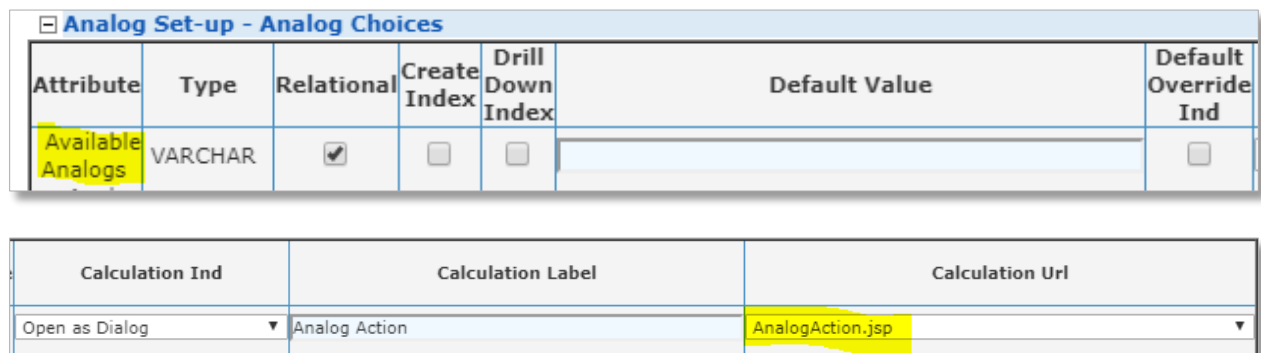
5. Now that we have set up the JSP and HTML files in both folders, we can access the attribute properties to point to those files in the Enable UI and the Enable 2020 UI.

5.2 Point to the Custom Action JSP/HTML

1. Using the server's Enable UI, in the **Feature Bar**, highlight the name of the repository that contains the attribute.
2. Click the arrow to the right of the repository's name to open the action menu, select **Edit**, and select **Attribute Properties**.



3. Navigate to the attribute the custom action will be applied to. (In this example we use Available Analogs.)

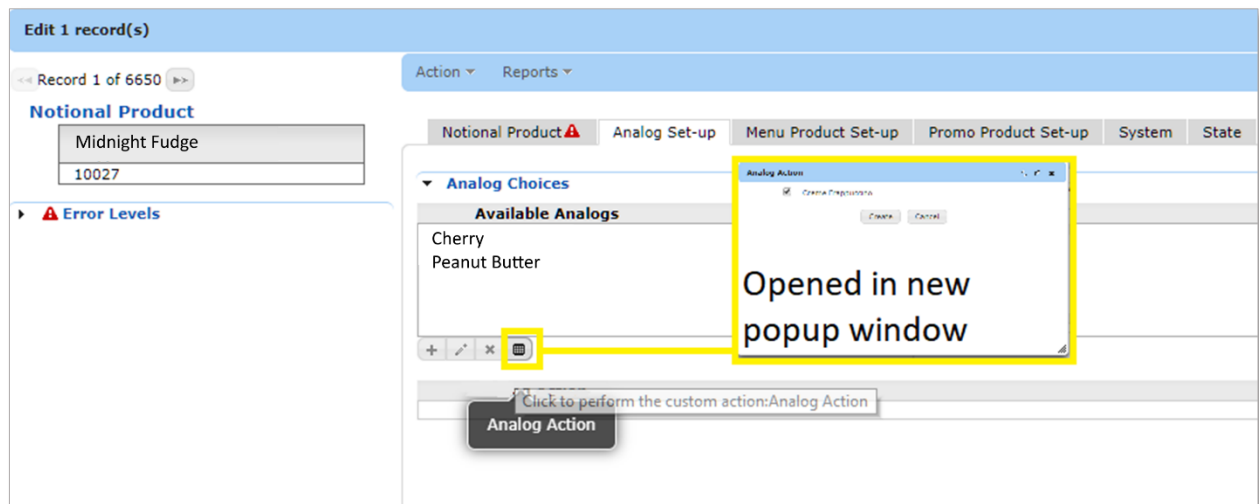


4. Edit the following settings as desired:
 - **Calculation Ind:** The action to occur when you click the custom action button (in this example it will open a new window).
 - **Calculation Label:** The label for the action.
 - **Calculation Url:** This is the file name of the HTML/JSP we created and placed inside the folders as described above.
5. When you have finished editing, click the **Save** button and close the window.

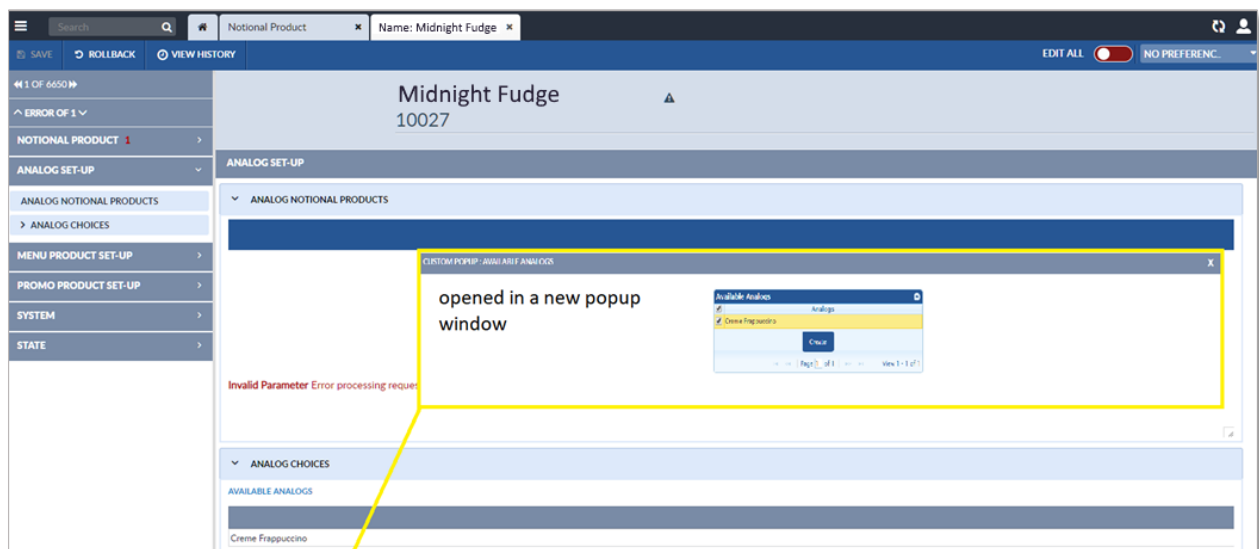
5.3 Check the Attribute Set Up

Now we will check that the attribute's custom action has been defined correctly in both the Enable UI and Enable 2020 UI.

Open a record that displays the attribute on which you have created the custom action. When you click the button, it should open your HTML/JSP page.



Testing the Custom Action in the Enable UI



Testing the Custom Action in the Enable 2020 UI

6 Write the HTML/JSP for the Enable 2020 UI

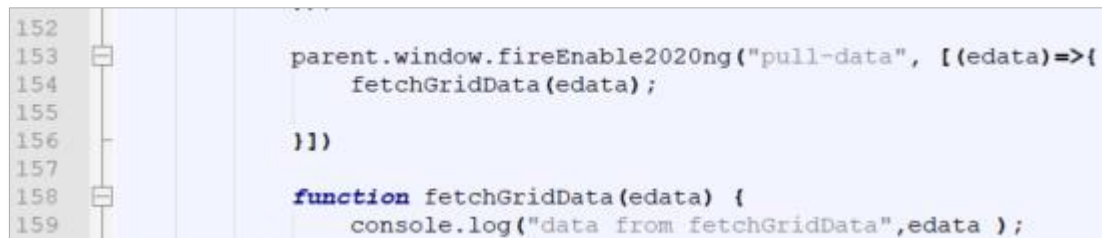
The HTML page displayed by the Enable 2020 UI is in:

\\<server>\d\$\Enterworks\enable2020\custom. It is composed of basic HTML elements combined with JavaScript.

6.1 Pull Data

In order to pull data from the item you are performing the action on, you need to use the `fireEnable2020ng` function.

Enable 2020 UI



```
152  
153  
154  
155  
156  
157  
158  
159  
parent.window.fireEnable2020ng("pull-data", [(edata)=>{  
    fetchGridData(edata);  
}])  
  
function fetchGridData(edata) {  
    console.log("data from fetchGridData", edata);
```

The fireEnable2020ng Function

The `fireEnable2020ng` function pulls data from the item you are performing the action on and dumps it into an `itemDataMap` (see [itemDataMap](#)). You can then use `itemDataMap` in combination with Enable's [Rest API Calls](#) to manipulate, create, and edit data.

6.2 Rest API Calls

Below is an example of how to incorporate a REST API call inside JavaScript:

```
278 function search(value){
279     var searchBody = {
280         "repositoryId": repoId,
281         "searchConfigList": [
282             {
283                 "repositoryId": repoId,
284                 "searchConfigAttrs": [
285                     {
286                         "defaultValue": value,
287                         "searchTypeCode": 10,
288                         "attributeId": newNameAttrId
289                     }
290                 ]
291             }
292         ]
293     };
294     searchBody = JSON.stringify(searchBody);
295     $.ajax({
296         url: "/webcm/rest/api/items/search",
297         dataType: 'json',
298         async: false,
299         type: 'post',
300         contentType: 'application/json',
301         data: searchBody,
302         success: function (result) {
303             if (result.data.searchTotal>0){
304                 found = true;
305             }
306         }
307     });
308 }
309
310
311
```

1Using AJAX and the REST Search API Call to Look for the Value Specified in a Repository

There are other ways to make HTTP requests in JavaScript, so you should figure out which works best for the custom action you are creating.

6.3 Classic UI HTML/JSP

In the Enable UI, you will notice that the JSP is written differently. To see examples/templates of how to write them please navigate to:

```
\\<server_name>\d$\Enterworks\EnableServer\tomcat\webapps\webcm\custom
```

7 Appendix

7.1 itemDataMap

```

1. {tempItemDataMap: {...}, itemThing: {...}, attr: {...}}
  1. attr:
    1. ALL_LANGUAGES: {Doc: {...}}
    2. ASSOC_GROUP_ID: null
    3. ATTRIBUTE_PRECISION: 0
    4. ATTRIBUTE_SCALE: 0
    5. ATTRIBUTE_SIZE: 200
    6. ATTR_GROUP_ID: 11258
    7. ATTR_ID: 12862
    8. ATTR_SECURITY_OWNER_IND: 0
    9. BUS_KEY_DUPS_IND: null
    10. BUS_KEY_GROUP_ID: null
    11. BUS_KEY_SEQ_NUM: null
    12. CODE_SET_ID: null
    13. COLUMN_DEFAULT: null
    14. CUSTOM_WIDGET_DETAILS: "select
    15.   ↵ analog.HierarchyNodeName as CODE
    16.   ↵ ,analog.HierarchyNodeName as DESCRIPTION
    17.   ↵ from Hierarchy_Property_Staging hero
    18.   ↵   join Hierarchy_Property_Relationship_Staging rel on
    19.   ↵   join Hierarchy_Property_Staging analog on
    20.   ↵ where hero.HierarchyName = 'enterprise product hierarchy'
    21.   ↵ and rel.RelationshipType = 'valid analog' and hero.HierarchyNode
    22.   ↵ = '[F_1004450]'"
    23. CUSTOM_WIDGET_IND: 211
    24. DATA_TYPE: "VARCHAR"
    25. DECIMAL_PADDING_IND: 0
    26. DESCRIPTION: "The choice of valid beverage styles that can
    27.   be used to create notional products using the same flavor
    28.   platform."
    29. DYN_ASSOC_ID: null
    30. DYN_ATTR_IND: 0
    31. FORMAT_ATTR_ID: 1004900
    32. HIERARCHY_IND: 0
    33. INDEXED_IND: null
    34. MASTER_CORE_IND: 0
    35. MULTI_LANG_SUPPORT_IND: 0
    36. NAME: "Available Analogs"
    37. NULL_IND: 0
    38. PKEY_SEQ_NUM: 0
    39. PROFILE_ID: 10119
  
```

```

36.      RADIX: null
37.      RECLOCK: null
38.      REPEATABLE_IND: 1
39.      REPEAT_SEPARATOR: "|"
40.      RESTRICTED_NAME: "AvailableAnalog"
41.      RFAM: {ATTR_DEFAULT: null, CALCULATION_BUTTON_LABEL: "Analog
      Action", CALCULATION_IND: 6, CALCULATION_JSP: "AnalogAction.jsp",
      CREATE_INDEX_IND: 0, ...}
42.      SEQUENCE_NUM: 13
43.      SEQ_GEN_IND: 0
44.      SPECIAL_FUNCTION_IND: null
45.      STATIC_IND: 1
46.      UUID: null
47.      VALIDATION_LEVEL_IND: null
48.      XPATH_ATTR: null
49.      attrGroupId: 11258
50.      destroyRequest: true
51.      isRestrictedHierarchy: false
52.      isTaxonomy: false
53.      name: "Available Analogs"
54.      systemName: "F_1004900"
55.      __proto__: Object
2. itemThing:
  1. allAttributes: {1004446: {...}, 1004448: {...}, 1004449: {...}, 1004450
    : {...}, 1004451: {...}, 1004452: {...}, 1004453: {...}, 1004455: {...}, 10
    04457: {...}, 1004458: {...}, 1004464: {...}, 1004466: {...}, 1004473: {...
    }, 1004519: {...}, 1004583: {...}, 1004585: {...}, 1004586: {...}, 100458
    7: {...}, 1004589: {...}, 1004657: {...}, 1004677: {...}, 1004692: {...}, 1
    004895: {...}, 1004899: {...}, 1004900: {...}, 1004913: {...}, 1004914: {...
    }, 1004915: {...}, 1004931: {...}, 1004976: {...}, 1004982: {...}, 10049
    86: {...}, 1004987: {...}, 1004988: {...}, 1004989: {...}, 1004990: {...},
    1004991: {...}, 1004992: {...}, 1004993: {...}, 1004994: {...}, 1004995:
    {...}, 1004996: {...}, 1004997: {...}, 1004998: {...}, 1004999: {...}, 1005
    000: {...}, 1005001: {...}, 1005002: {...}, 1005004: {...}, 1005101: {...}}
  2. allCustomAttributes: {1004900: {...}, 1004986: {...}, 1005002: {...}, 1
    005004: {...}, 1005101: {...}}
  3. attrTabs: (6) [{...}, {...}, {...}, {...}, {...}, {...}]
  4. bPendingRequestAttr: []
  5. childLinks: []
  6. errorFilter: [{...}]
  7. hierarchyList: (3) [{...}, {...}, {...}]
  8. highlightFilter: (44) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...
    }, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...
    }, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...
    }, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...
    }, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}
  9. item: {itemId: 6169737, stagingItemId: 0, messageId: null, transa
    ctionId: null, modifyAction: "update", ...}
  10.      itemId: "6169737"

```

```

11.     parentLinks: (4) [{...}, {...}, {...}, {...}]
12.     profile: {profileId: 10119, name: "Notional
Product", description: "", currencyFormat: null, creationDatetime
: "2019-01-02T03:16:43.573", ...}
13.     repository: {ALL_LANGUAGES: {...}, AUTO_GTIN_IND: 1, AUTO_PUB
LISH_IND: 0, AUTO_PUBLISH_INTERVAL: null, AUTO_PUBLISH_SEED_TIME:
null, ...}
14.     repositoryId: "10134"
15.     __proto__: Object
3. tempItemDataMap:
  1. F_1004446: "10027"
  2. F_1004448: "Peppermint Mocha Coffee Frappuccino"
  3. F_1004449: "test update
  4. ↵"
  5. F_1004450: "1300"
  6. F_1004451: null
  7. F_1004452: null
  8. F_1004453: null
  9. F_1004455: "New"
 10.   F_1004457: "04/24/2019"
 11.   F_1004458: "Signature"
 12.   F_1004464: null
 13.   F_1004473: "23"
 14.   F_1004519: null
 15.   F_1004583: null
 16.   F_1004586: "Signature"
 17.   F_1004657: "Regular"
 18.   F_1004692: "Peppermint Mocha"
 19.   F_1004895: "1"
 20.   F_1004899: "EPH"
 21.   F_1004900: "Creme Frappuccino"
 22.   F_1004913: "Coffee Frappuccino"
 23.   F_1004914: "Frappuccino"
 24.   F_1004915: "Beverage"
 25.   F_1004931: "0"
 26.   F_1004982: "10000"
 27.   F_1004986: null
 28.   F_1004987: null
 29.   F_1004988: null
 30.   F_1004989: null
 31.   F_1004990: null
 32.   F_1004991: null
 33.   F_1004992: null
 34.   F_1004993: null
 35.   F_1004994: null
 36.   F_1004995: null
 37.   F_1004996: null
 38.   F_1004997: null
 39.   F_1004998: null

```

```
40.      F_1004999: null
41.      F_1005000: null
42.      F_1005001: null
43.      F_1005002: null
44.      F_1005004: null
45.      F_1005101: null
46.      __proto__: Object
4.  __proto__: Object
```