



**SERENA®**  
**BUSINESS MANAGER**  
**JSON API Guide**

---

Copyright © 2007–2016 Serena Software, Inc. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Serena. Any reproduction of such software product user documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification. This document contains proprietary and confidential information, and no reproduction or dissemination of any information contained herein is allowed without the express permission of Serena Software.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

License and copyright information for 3rd party software included in this release can be found on the product's news page at <http://support.serena.com/ProductNews/default.aspx> and may also be found as part of the software download available at <http://support.serena.com>.

## **Trademarks**

Serena, Dimensions, ChangeMan, Comparex, and StarTool are registered trademarks of Serena Software, Inc. The Serena logo, PVCS, TeamTrack, License Manager and Composer are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

## **U.S. Government Rights**

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 2345 NW Amberbrook Drive, Suite 200, Hillsboro, OR 97006.

Part number: Product version: 11.2

Publication date: 2016-11-22

---

# Table of Contents

Overview .....	4
Access and Authentication .....	4
Requesting a Security Token .....	5
Reference.....	7
GET Functions .....	7
GetItem .....	7
GetItemsByItemID .....	9
GetItemsByListingReport .....	12
GetItemsByReportFilter .....	16
GetItemTransitions .....	19
GetVersion .....	20
POST Functions .....	21
FinishTransition .....	21
StartSubmitToAux .....	24
StartSubmitToProject .....	26
StartTransition .....	29
SubmitToAux.....	32
SubmitToProject .....	35
JSON API Options .....	37
ItemOptions .....	37
Transition .....	39

## Overview

This document describes how to use the SBM JSON API, a product of Serena Software, Inc. SBM is a Web-architected change request and process management solution that empowers application development teams to improve communication and development processes across the enterprise.

The SBM JSON API provides a set of RESTful services that enable you to perform CRUD (create, read, update, and delete) operations against SBM items via GET and POST calls.

For example:

- You can send a JSON API POST call from JavaScript on a custom form to update an item.
- You can create a REST data source in Composer and initiate a GET call against a separate SBM instance to return external items and populate a Kanban board in Serena Work Center.
- You can work with SBM items by invoking GET and POST calls from any external application that is capable of sending RESTful calls.

## Audience and Scope

This document is intended for experienced developers interested in integrating their products with SBM.



**Important:** Support for development efforts writing REST Web services is provided by Professional Services. Questions regarding use of API functions as documented in the following sections are handled by customer support.

## Access and Authentication

This section describes how to access all of the JSON API functions.

To access the API directly, use the following format:

```
http://AEserverName/jsonapi/getitem/1000/2
```

Alternatively, start with any of the following addresses:

```
http://AEserverName/tmtrack/tmtrack.dll?  
http://AEserverName/workcenter/tmtrack.dll?  
http://AEserverName/workcenter/tmtrack.dll?shell=swc&
```

and append the function call (JSON\_Func=) and any required parameters (JSON\_P1=, JSON\_P2=):

```
JSONPage&command=jsonapi&JSON_Func=getitem&JSON_P1=1000&JSON_P2=2
```

---

Note that the order of the parameters on the URL is important. For example, the `GetItem` call fails if you have the table ID parameter value in the item ID parameter and vice versa.



**Tip:** If you are calling a function via JavaScript on a custom form, begin the URL with `tmtrack.dll?JSONPage&command=`. This ensures the call will succeed in both the classic interface and in Work Center.

Every JSON API call contains a result in the response. For every call that is successfully authenticated, the result contains a `type` of either `OK` or `ERROR`. Every `ERROR` result has an English message after `msg` and may also contain a localizable message in `msgLoc`.

```
"result":{"type":"OK","msg":""}
"result":{"type":"ERROR","msg":"Could not read specified record by ID"}
"result":{"type":"ERROR","msg":"Could not read specified record by ID",
→"msgLoc":"The item could not be found."}
```

All functions are accessible from the browser using existing authentication methods such as Windows Authentication or Single Sign-On (SSO). You can use the `TokenService` call to acquire and send an SSO token in the HTTP header for SBM server configurations that use SSO. For details, refer to [Requesting a Security Token \[page 5\]](#).

If authentication fails, or the required SSO token is not found, the `authentication type` is returned with the message `Authentication failed`:

```
{
"request": {
"url": "http://AE/tmtrack/tmtrack.dll?JSONPage&command=jsonapi&JSON_Func=getVersion&"
}
"error": {
"type": "authentication"
"msg": "Authentication failed"
}
"sso": {
"idp": "http://SSOserverName:8085/idp/services/Trust"
"msg": ""
"code": ""
}
}
```

All functions provided by the JSON API honor SBM user privileges; additional privileges are not required to access the API functions.

## Requesting a Security Token

For SBM systems that are configured to use Single Sign-On (SSO), you must provide a security token in the HTTP header with the JSON call. Upon successful authentication, the `TokenService` call returns a valid SSO SAML token that you can pass in subsequent JSON API calls, SOAP-based calls via the SBM Application Engine Web Services API, or other SBM interfaces that require a valid SAML token.

### To request a security token:

1. Invoke the `TokenService` POST call using the following URL (with your SSO server's host name):

`http://SSOHostName:8085/idp/services/rest/TokenService/`

2. Pass the following header information:

- **Content-Type:** `application/json;charset=UTF-8`
- **Accept:** `application/json`

3. Include authentication credentials in the payload:

```
{"credentials": { "username" : "bill", "password":"mypassword"}}
```

For example:

```
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 56
Source message
```

```
POST /idp/services/rest/TokenService/ HTTP/1.1
HOST: SSOHostName:8085
content-type: application/json;charset=UTF-8
accept: application/json
content-length: 56
```

```
{"credentials": { "username" : "bill", "password":"mypassword"}}
```

If authentication succeeds, a response message is returned:

```
{
"status": "OK"
"lifetime": {
"created": 1467228623885
"expires": 1467257423885
}
"token": {
"tokenType": "SSO"
"encoding": "base64"
"value": "PHNhbWw6QXNzZXJ0...pBc3NlcnRpb24="
```

The token information is in the base64-encoded `value` string. You now have a valid security token that you can pass with `ALFSSOAuthNToken` in the HTTP header in subsequent JSON API calls or as part of the WS-trust envelope for SOAP-based calls made via the SBM Application Engine Web Services API.

The following example shows how to pass the token in `ALFSSOAuthNToken` with the `GetVersion` call:

```
GET /jsonapi/GetVersion HTTP/1.1
HOST: SSOHostName
alfssoauthntoken: PHNhbWw6QXNzZXJ0...<base64 encoded string>...dGlvbj4=
cookie: authrequired=
```

## Related Topics

- 
- [GET Functions \[page 7\]](#)
  - [POST Functions \[page 21\]](#)

## Reference

The functions described in this section are divided into the following categories:

- [GET Functions \[page 7\]](#)
- [POST Functions \[page 21\]](#)

## GET Functions

- [GetItem \[page 7\]](#)
- [GetItemsByItemID \[page 9\]](#)
- [GetItemsByListingReport \[page 12\]](#)
- [GetItemsByReportFilter \[page 16\]](#)
- [GetItemTransitions \[page 19\]](#)
- [GetVersion \[page 20\]](#)

## GetItem

Gets a primary or auxiliary item.

### Parameters

Name	Type	Description
<code>GetItem</code>	Function	Returns a primary or auxiliary item.
<code>table</code>	<ul style="list-style-type: none"><li>• ID (integer)</li><li>• UUID (string)</li><li>• dbname (string)</li><li>• name (string)</li></ul>	The primary or auxiliary table identifier.
<code>item</code>	<ul style="list-style-type: none"><li>• ID (integer)</li><li>• UUID (string)</li></ul>	The item identifier.

## Post Data

You can optionally use the GetItem function in an HTTP POST and include one or more of the following objects:

Name	Type	Description
itemOptions (optional)	ItemOptions <a href="#">[page 37]</a>	Use itemOptions to limit the response data. When itemOptions are included, only the specified fields, plus fields required by the transition, are sent in the response.
getTransitions (optional)	boolean	Include getTransitions to return a list of transitions that are currently available for this item.

## Example

URL:

```
http://AEServerName/jsonapi/GetItem/1000/109
```

Post data:

```
{fixedFields: false, includeNotes: true,  
→fields: [{dbname:"TITLE"}, {dbname: "FUNCTIONAL_AREA"}]}
```

Response:

```
{  
  "item": {  
    "id": {  
      "id": 109,  
      "uuid": "c1e62a05-5dd1-46e7-9cdc-b91518893b23",  
      "itemIdPrefixed": "BUG000173",  
      "itemId": "000173",  
      "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...&TableId=1000"  
    },  
    "table": {  
      "id": 1000,  
      "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",  
      "name": "Issues",  
      "dbName": "UBG_ISSUES"  
    },  
    "fieldMetadata": [  
      {  
        "id": 54,  
        "name": "Title",  
        "dbname": "TITLE",  
        "syscode": 4,  
        "type": "text",  
        "renderHtml": false,  
      }  
    ]  
  }  
}
```



```

    "password": false,
    "maxLength": 80
  },
  {
    "id": 75,
    "name": "Functional Area",
    "dbname": "FUNCTIONAL_AREA",
    "syscode": 0,
    "type": "selection",
    "renderHtml": false,
    "isMultiSelect": false
  }
],
"fields": {
  "TITLE": {
    "value": "Image Builder crashes my system."
  },
  "FUNCTIONAL_AREA": {
    "id": 30,
    "name": "User Interface"
  }
},
"notes": []
},
"result": {
  "type": "OK",
  "msg": ""
}
}
}

```

## Comments

Because the `TS_ID` is subject to change in a promotion context, you may need to update the `table` and `item` parameters with the `TS_ID` from the target environment after the promote is finished.

## GetItemsByItemID

Gets multiple items matching the specified prefix and ID.

### Parameters

Name	Type	Description
GetItemsByItemId	Function	Gets multiple items matching the prefix and ID that are passed. Item IDs are not unique, so this query returns an array of results. Unlike the report-based calls, you cannot request a second page of results with this query. Results are sorted by the Item ID field.

table	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• dbname (string)</li> <li>• name (string)</li> </ul>	<p>The primary or auxiliary table identifier.</p> <p>Note that <code>name</code> is case sensitive and can be changed in SBM Composer and deployed.</p>
itemId	<ul style="list-style-type: none"> <li>• ID (string)</li> </ul>	<p>The item ID. For example: BUG2, ENH000079, *, %, BUG, BUG*, BUG*2, BUG*2*.</p>

## Usage

In the direct access URL, you can specify the item ID as BUG2 or BUG000002:

```
http://AEServerName/jsonapi/getitemsbyitemid/Issues/BUG2?pagesize=2
```

Note that you cannot use \* or % in the path part of the URL. However, in the longer URL format, you can specify either wildcard.

For example:

```
JSONPage&command=jsonapi&JSON_Func=getitemsbyitemid&JSON_P1=1000&JSON_P2=BUG*2
```

## Post Data

You can optionally use the GetItemsByItemID function in an HTTP POST and include one or more of the following objects:

Name	Type	Description
itemOptions (optional)	ItemOptions <a href="#">[page 37]</a>	Use <code>itemOptions</code> to limit the response data. When <code>itemOptions</code> are included, only the specified fields are sent in the response.
pagesize (optional)	integer	Add <code>&amp;pagesize=100</code> to the URL to override the user's page size preference. For example, <code>&amp;pagesize=1</code> ensures that only the first item found is returned.

## Example

URL:

---

http://AEServerName/jsonapi/GetItemsByItemID/Issues/BUG?pagesize=2

**Post data:**

```
{fixedFields: false, fields: [{dbname:"TITLE"}, {dbname: "FUNCTIONAL_AREA"}]}
```

**Response:**

```
{
  "searchResults": {
    "totalItems": 69,
    "pageSize": 2
  },
  "table": {
    "id": 1000,
    "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",
    "name": "Issues",
    "dbName": "UBG_ISSUES"
  },
  "fieldMetadata": [
    {
      "id": 54,
      "name": "Title",
      "dbname": "TITLE",
      "syscode": 4,
      "type": "text",
      "renderHtml": false,
      "password": false,
      "maxLength": 80
    },
    {
      "id": 75,
      "name": "Functional Area",
      "dbname": "FUNCTIONAL_AREA",
      "syscode": 0,
      "type": "selection",
      "renderHtml": false,
      "isMultiSelect": false
    }
  ],
  "items": [
    {
      "id": {
        "id": 2,
        "uuid": "46ad4b77-4c8e-445d-b6d8-e84b71cfb8ab",
        "itemIdPrefixed": "BUG000002",
        "itemId": "000002",
        "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1000"
      },
      "fields": {
        "TITLE": {
          "value": "Upgrade from 4.0 does not work correctly"
        },
        "FUNCTIONAL_AREA": {

```

```

        "id": 0,
        "name": "(None)"
    }
}
},
{
  "id": {
    "id": 13,
    "uuid": "9506eb9c-0822-4944-b023-5142696c64d1",
    "itemIdPrefixed": "BUG000028",
    "itemId": "000028",
    "url": "http://AEserverName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1000"
  },
  "fields": {
    "TITLE": {
      "value": "404 page not found error"
    },
    "FUNCTIONAL_AREA": {
      "id": 0,
      "name": "(None)"
    }
  }
}
},
],
"result": {
  "type": "OK",
  "msg": ""
}
}
}

```

## Comments

Because the TS\_ID is subject to change in a promotion context, you may need to update the `table` parameter with the TS\_ID from the target environment after the promote is finished.

## GetItemsByListingReport

Gets primary or auxiliary items from a listing report.

### Parameters

There are two ways to specify the report. Parameters differ depending on the option you use.

Option 1 (`report`):

Name	Type	Description
GetitemsByListingReport	Function	Returns primary or auxiliary items from a specified listing report.

report	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> </ul>	The listing report identifier.
--------	---	--------------------------------

Option 2 (table and reportReferenceName):

Name	Type	Description
GetitemsByListingReport	Function	Returns a primary or auxiliary item.
table	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• dbname (string)</li> <li>• name (string)</li> </ul>	<p>The table identifier.</p> <p>For example:</p> <pre>UBG_ISSUES</pre>
reportReferenceName	string	<p>The reference name for the report, used in combination with the table identifier.</p> <p>For example:</p> <pre>IDM_all_active_crs</pre>

## Post Data

You can optionally use the GetItemsByListingReport function in an HTTP POST and include one or more of the following objects:

Name	Type	Description
itemOptions (optional)	<a href="#">ItemOptions [page 37]</a>	<p>Use <code>itemOptions</code> to limit the response data.</p> <p>When <code>itemOptions</code> are included, only the specified fields, plus fields required by the transition, are sent in the response.</p>

hasruntimeparams (optional)	integer	Use <code>hasruntimeparams</code> with query-at-runtime reports to fill in the blanks in the report filter. For example:  example: <code>&amp;hasruntimeparams=1&amp;f80v1=0</code>
pagesize (optional)	integer	Report results honor the user's page size preferences. You can override this setting using <code>pagesize</code> , but the results will still be limited to the maximum that is set in the system setting "MaxItemsPerPage". For example:  example: <code>&amp;pagesize=5000</code>
rptkey and recno (both optional)	integer	Reports cache the result set under the key value <code>rptkey</code> . A second page of report results can be requested by repeating the report call and appending <code>rptkey</code> and the desired starting index (0-based) to the URL. For example:  example: <code>&amp;rptkey=1312321&amp;recno=20</code>

## Example

### URL:

```
http://AEServerName/jsonapi/getitemsbylistingreport/106?pagesize=2
```

### Post data:

```
{fixedFields: false, fields: [{dbname:"TITLE"}, {dbname: "FUNCTIONAL_AREA"}]}
```

### Response:

```
{
  "report": {
    "id": 106,
    "uuid": "8949036c-c217-480b-9214-f831fda3155f",
    "name": "All Issues With Reject Count > 0",
    "comment": "",
    "url": "tmtrack.dll?ReportPage&Template=reports%2Flist&ReportId=106",
    "refURL": "tmtrack.dll?ReportPage&template=reports%2Flist&ReportUUID=
→8949036c-c217-480b-9214-f831fda3155f&embedded"
  },
  "reportResults": {
```

---

```

    "totalItems": 6,
    "pageSize": 2,
    "firstRecordIndex": 0,
    "rptkey": 1846002703
  },
  "table": {
    "id": 1000,
    "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",
    "name": "Issues",
    "dbName": "UBG_ISSUES"
  },
  "fieldMetadata": [
    {
      "id": 54,
      "name": "Title",
      "dbName": "TITLE",
      "syscode": 4,
      "type": "text",
      "renderHtml": false,
      "password": false,
      "maxLength": 80
    },
    {
      "id": 75,
      "name": "Functional Area",
      "dbName": "FUNCTIONAL_AREA",
      "syscode": 0,
      "type": "selection",
      "renderHtml": false,
      "isMultiSelect": false
    }
  ],
  "items": [
    {
      "id": {
        "id": 109,
        "uuid": "",
        "itemIdPrefixed": "BUG000173",
        "itemId": "000173",
        "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...view&TableId=1000"
      },
      "fields": {
        "TITLE": {
          "value": "Image Builder crashes my system."
        },
        "FUNCTIONAL_AREA": {
          "id": 30,
          "name": "User Interface"
        }
      }
    }
  ],
  {
    "id": {
      "id": 47,
      "uuid": "",

```

```

    "itemIdPrefixed": "BUG000084",
    "itemId": "000084",
    "url": "http://AEserverName:80/tmtrack/tmtrack.dll?...view&TableId=1000"
  },
  "fields": {
    "TITLE": {
      "value": "Program shuts down"
    },
    "FUNCTIONAL_AREA": {
      "id": 30,
      "name": "User Interface"
    }
  }
}
],
"result": {
  "type": "OK",
  "msg": ""
}
}

```

## Comments

Because the TS\_ID is subject to change in a promotion context, you may need to update the `report` and `table` parameters with the TS\_ID from the target environment after the promote is finished.

## GetItemsByReportFilter

Gets multiple items using a report filter.

### Parameters

Name	Type	Description
GetitemsByReportFilter	Function	Gets multiple items using a report filter to identify which items to return.
reportFilter	<ul style="list-style-type: none"> <li>ID (integer)</li> <li>UUID (string)</li> <li>name (string)</li> </ul>	The report filter identifier.

### Post Data

You can optionally use the `GetItemsByReportFilter` function in an HTTP POST and include one or more of the following objects:



Name	Type	Description
itemOptions (optional)	ItemOptions <a href="#">[page 37]</a>	Use itemOptions to limit the response data. When itemOptions are included, only the specified fields are sent in the response.

## Example

URL:

`http://AEServerName/jsonapi/getitemsbyreportfilter/1?pagesize=2`

Post data:

```
{fixedFields: false, includeNotes: true,
→fields: [{dbname:"TITLE"}, {dbname: "FUNCTIONAL_AREA"}]}
```

Response:

```
{
  "reportFilter": {
    "id": 1,
    "uuid": "ac956c79-3784-4c92-bdaf-d0f6a98f1ba8",
    "name": "IDM report filter"
  },
  "reportResults": {
    "totalItems": 60,
    "pageSize": 2,
    "firstRecordIndex": 0,
    "rptkey": 284626196
  },
  "table": {
    "id": 1000,
    "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",
    "name": "Issues",
    "dbName": "UBG_ISSUES"
  },
  "fieldMetadata": [
    {
      "id": 54,
      "name": "Title",
      "dbname": "TITLE",
      "syscode": 4,
      "type": "text",
      "renderHtml": false,
      "password": false,
      "maxLength": 80
    },
    {
      "id": 75,
      "name": "Functional Area",
      "dbname": "FUNCTIONAL_AREA",
```

```
    "syscode": 0,
    "type": "selection",
    "renderHtml": false,
    "isMultiSelect": false
  }
],
"items": [
  {
    "id": {
      "id": 2,
      "uuid": "",
      "itemIdPrefixed": "BUG000002",
      "itemId": "000002",
      "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1000"
    },
    "fields": {
      "TITLE": {
        "value": "Upgrade from 4.0 does not work correctly"
      },
      "FUNCTIONAL_AREA": {
        "id": 0,
        "name": "(None)"
      }
    }
  },
  {
    "id": {
      "id": 3,
      "uuid": "",
      "itemIdPrefixed": "ENH000005",
      "itemId": "000005",
      "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1000"
    },
    "fields": {
      "TITLE": {
        "value": "Change icons to more modern look"
      },
      "FUNCTIONAL_AREA": {
        "id": 0,
        "name": "(None)"
      }
    }
  }
],
"result": {
  "type": "OK",
  "msg": ""
}
}
```

## Comments

Because the TS\_ID is subject to change in a promotion context, you may need to update the `reportFilter` parameter with the TS\_ID from the target environment after the promote is finished.

---

## GetItemTransitions

Gets available transitions for a primary item.

### Parameters

Name	Type	Description
Getitemtransitions	Function	Returns a list of available transitions for an item. The calling user's privileges are considered when returning the list of transitions.  Use GetItemTransitions to identify transitions for use with other calls.
table	<ul style="list-style-type: none"><li>• ID (integer)</li><li>• UUID (string)</li><li>• dbname (string)</li><li>• name (string)</li></ul>	The primary table identifier.
item	<ul style="list-style-type: none"><li>• ID (integer)</li><li>• UUID (string)</li></ul>	The item identifier.

### Post Data

None.

### Example

URL:

```
http://AEServerName/jsonapi/getitemtransitions/1000/2
```

Response:

```
{
  "transitions": [
    {
      "id": 18,
```

```
    "name": "Re-Open",
    "uuid": "e44d9bfe-6d54-426e-a1ac-4711a09b6243",
    "type": "regular",
    "isQuick": false
  },
  {
    "id": 1,
    "name": "Update",
    "uuid": "update",
    "internalName": "UPDATE",
    "type": "update",
    "isQuick": false
  },
  {
    "id": 2,
    "name": "Delete",
    "uuid": "delete",
    "internalName": "DELETE",
    "type": "delete",
    "isQuick": false
  }
],
"result": {
  "type": "OK",
  "msg": ""
}
}
```

## Comments

Because the TS\_ID is subject to change in a promotion context, you may need to update the `table` and `item` parameters with the TS\_ID from the target environment after the promote is finished.

## GetVersion

Gets the current Application Engine version.

### Parameters

Name	Type	Description
GetVersion	Function	Returns the version of the target Application Engine.

### Post Data

None.

### Example

URL:

```
http://AEServerName/jsonapi/getversion
```

---

Response:

```
{
  "version": "Version 11.01.00.00.354",
  "result": {
    "type": "OK",
    "msg": ""
  }
}
```

## Comments

None.

## POST Functions

- [FinishTransition \[page 21\]](#)
- [StartSubmitToAux \[page 24\]](#)
- [StartSubmitToProject \[page 26\]](#)
- [StartTransition \[page 29\]](#)
- [SubmitToAux \[page 32\]](#)
- [SubmitToProject \[page 35\]](#)


## FinishTransition

Completes a transition on an item.

The specified item will get a recordlock if the table uses record-locking. This record-lock is passed to the FinishTransition call.

### Parameters

Name	Type	Description
FinishTransition	Function	Completes a transition on an item. You can send new field values with the post data using the <code>transition</code> object.

table	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• dbname (string)</li> <li>• name (string)</li> </ul>	The table identifier.
item	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> </ul>	The item identifier.
transition	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• internal name (string)</li> </ul>	<p>The transition identifier to use. If 0, a valid Update transition is used (if found).</p> <p> <b>Tip:</b> Run GetItemTransitions first to determine the transition identifier that you will specify.</p> <p>Not all transitions have an internal name. Those that do return the internal name as <code>dbname</code> in the GetItemTransitions response.</p>
record-lock ID	ID (integer)	<p>Use one of the following:</p> <ul style="list-style-type: none"> <li>• The ID value from the StartTransition response</li> <li>• 0 if the table does not use record locking</li> <li>• -1 if you did not invoke StartTransition and you want to run the transition without first getting the current field values of the item.</li> </ul>

## Post Data

Name	Type	Description
------	------	-------------

itemOptions (optional)	<a href="#">ItemOptions [page 37]</a>	Used to define the item's values that will be included in the response if the transition succeeds.  When <code>itemOptions</code> are included, only the specified fields are sent in the response.
breakLock (optional)	boolean	Allows the JSON API to steal the record lock from any user who might have one on this item, so that this call should not fail due to a user having the record locked. However, this causes any user who has a transition form open on this item to get an error when he or she tries to commit the transition.
transition (optional)	object	Use the <code>transition</code> object to provide new field values for the transition. Only provide values for fields you want to change.

## Example

URL:

```
http://AEserverName/jsonapi/finishTransition/1000/3/18/-1
```

Post data:

```
{fixedFields: false, fields: [{dbname: "STATUS_LOG"}, {dbname: "TESTER"}],
→transition: {"STATUS_LOG": "Reopen issue", "TESTER": "7"}}
```

Response:

```
{
  "item": {
    "id": {
      "id": 3,
      "uuid": "c2379ef5-6f2d-4e21-924f-5c2aa2a0382e",
      "itemIdPrefixed": "ENH000005",
      "itemId": "000005",
      "url": "http://AEserverName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1000"
    },
    "table": {
      "id": 1000,
      "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",
      "name": "Issues",
      "dbName": "UBG_ISSUES"
    },
    "fieldMetadata": [
      {
        "id": 65,
        "name": "Status Log",
        "dbname": "STATUS_LOG",
        "syscode": 0,

```

```

    "type": "journal",
    "renderHtml": false,
    "appendOnly": true,
    "sortNewestFirst": false
  },
  {
    "id": 63,
    "name": "Tester",
    "dbname": "TESTER",
    "syscode": 0,
    "type": "user",
    "renderHtml": false,
    "isMultiSelect": false
  }
],
"fields": {
  "STATUS_LOG": {
    "value": "07/13/2016 03:49:48 PM - Bill Admin:\r\n\nReopen issue\r\n"
  },
  "TESTER": {
    "id": 3,
    "name": "Chris Tester"
  }
}
},
"result": {
  "type": "OK",
  "msg": ""
}
}

```

## Comments

None.

## StartSubmitToAux

Starts a submit to a specified auxiliary table.

## Parameters

Name	Type	Description
startsubmittoaux	Function	Starts a submit to a specified auxiliary table. The main purpose of this function is to get the item's current field values, as well as information about which fields are required or read-only.



table	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• internalname (string)</li> <li>• name (string)</li> </ul>	The table identifier.
-------	---	-----------------------

## Post Data

Name	Type	Description
itemOptions (optional)	<a href="#">ItemOptions [page 37]</a>	Use itemOptions to limit the response data.  When itemOptions are not included, all fields are sent in the response. When itemOptions are included, only the specified fields, plus fields required by the transition, are sent in the response.

## Example

URL:

`http://AEServerName/jsonapi/startsubmittoaux/UBG_IDM_CONFIGURATIONS`

Post data:

```
{fixedFields: false, fields: [{dbname:"TITLE"}, {dbname: "FIELD_NAME"}]}
```

Response:

```
{
  "startSubmitToAux": {
    "item": {
      "table": {
        "id": 1004,
        "uuid": "dcfead12-5fa4-471d-af37-7f9a36dee2c3",
        "name": "IDM Configurations",
        "dbName": "UBG_IDM_CONFIGURATIONS"
      },
      "fieldMetadata": [
        {
          "id": 152,
          "name": "Application",
          "dbname": "TITLE",
          "syscode": 4,
          "type": "text",
          "required": true,
          "readonly": false,

```

```

    "renderHtml": false,
    "sectionId": 1,
    "helpText": "Name of the application the configuration record pertains to",
    "password": false,
    "maxLength": 80
  },
  {
    "id": 153,
    "name": "Field Name",
    "dbname": "FIELD_NAME",
    "syscode": 0,
    "type": "text",
    "required": true,
    "readonly": false,
    "renderHtml": false,
    "sectionId": 1,
    "helpText": "The field name of the configuration data",
    "password": false,
    "maxLength": 80
  }
],
"fields": {
  "TITLE": {
    "value": ""
  },
  "FIELD_NAME": {
    "value": ""
  }
}
}
},
"result": {
  "type": "OK",
  "msg": ""
}
}

```

### Comments

None.

## StartSubmitToProject

Starts a submit to a specified project.

### Parameters

Name	Type	Description
------	------	-------------

startsubmittoproject	Function	Starts a submit to a specified project. The main purpose of this function is to get the item's current field values, as well as information about which fields are required or read-only.
project	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• internalname (string)</li> </ul>	The project identifier.

## Post Data

Name	Type	Description
itemOptions (optional)	<a href="#">ItemOptions [page 37]</a>	Use <code>itemOptions</code> to limit the response data. When <code>itemOptions</code> are not included, all fields are sent in the response. When <code>itemOptions</code> are included, only the specified fields, plus fields required by the transition, are sent in the response.

## Example

URL:

`http://AEServerName/jsonapi/startsubmittoproject/UBG_ISSUES.ANIMATION_PRO`

Post data:

```
{fixedFields: false, includeNotes: true,
→fields: [{dbname:"TITLE"}, {dbname: "REPRODUCIBLE_"}]}
```

Response:

```
{
  "startSubmitToProject": {
    "transition": {
      "id": 3,
      "name": "Submit",
      "uuid": "a78f0a30-1305-46c2-b661-df8219c105b2",
      "type": "regular",
      "isQuick": false,
      "noteRequired": false,
      "urlRequired": false,
      "fileRequired": false,
      "itemRequired": false
    }
  }
}
```

```
},
"item": {
  "table": {
    "id": 1000,
    "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",
    "name": "Issues",
    "dbName": "UBG_ISSUES"
  },
  "fieldMetadata": [
    {
      "id": 54,
      "name": "Title",
      "dbName": "TITLE",
      "syscode": 4,
      "type": "text",
      "required": true,
      "readonly": false,
      "renderHtml": false,
      "sectionId": 1,
      "helpText": "",
      "password": false,
      "maxLength": 80
    },
    {
      "id": 70,
      "name": "Reproducible",
      "dbName": "REPRODUCIBLE_",
      "syscode": 0,
      "type": "binary",
      "required": false,
      "readonly": false,
      "renderHtml": false,
      "sectionId": 2,
      "helpText": "",
      "selections": [
        {
          "id": 0,
          "name": "No"
        },
        {
          "id": 1,
          "name": "Yes"
        }
      ]
    }
  ]
},
"fields": {
  "TITLE": {
    "value": ""
  },
  "REPRODUCIBLE_": {
    "id": 0,
    "name": "No"
  }
}
},
```

```
    "notes": []
  }
},
"result": {
  "type": "OK",
  "msg": ""
}
}
```

## Comments

None.


## StartTransition

Starts a transition on an item.

The specified item will get a recordlock if the table uses record-locking. This record-lock is passed to the FinishTransition call.

## Parameters

Name	Type	Description
StartTransition	Function	Starts a transition on an item. Use this function to lock the item and get current field values, as well as information about which fields are required or read-only in the transition.
table	<ul style="list-style-type: none"><li>ID (integer)</li><li>UUID (string)</li><li>dbname (string)</li><li>name (string)</li></ul>	The table identifier.
item	<ul style="list-style-type: none"><li>ID (integer)</li><li>UUID (string)</li></ul>	The item identifier.

transition	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• internal name (string)</li> </ul>	<p>The transition to start. If 0, a valid Update transition is used (if found).</p> <p> <b>Tip:</b> Run <code>GetItemTransitions</code> first to determine the transition identifier that you will specify.</p> <p>Not all transitions have an internal name. Those that do return the internal name as <code>dbname</code> in the <code>GetItemTransitions</code> response.</p>
------------	---	---

## Post Data

Name	Type	Description
itemOptions (optional)	ItemOptions [page 37]	<p>Use <code>itemOptions</code> to limit the response data.</p> <p>When <code>itemOptions</code> are not included, all fields are sent in the response. When <code>itemOptions</code> are included, only the specified fields, plus fields required by the transition, are sent in the response.</p>
breakLock (optional)	boolean	<p>Allows the JSON API to steal the record lock from any user who might have one on this item, so that this call should not fail due to a user having the record locked. However, this causes any user who has a transition form open on this item to get an error when he or she tries to commit the transition.</p>

## Example

URL:

```
http://AEServerName/jsonapi/startTransition/1000/2/18
```

Post data:

```
{fixedFields: false, fields: [{dbname: "FUNCTIONAL_AREA"}], breaklock: true}
```



**Note:** In this example, only Functional Area is returned to illustrate the response structure. You will typically want to return all fields to determine which fields are required for the transition.

Response:

```
{
  "startTransition": {
    "transition": {
      "id": 18,
      "name": "Re-Open",

```

---

```
"uuid": "e44d9bfe-6d54-426e-a1ac-4711a09b6243",
"type": "regular",
"isQuick": false,
"noteRequired": false,
"urlRequired": false,
"fileRequired": false,
"itemRequired": false
},
"item": {
  "id": {
    "id": 2,
    "uuid": "46ad4b77-4c8e-445d-b6d8-e84b71cfb8ab",
    "itemIdPrefixed": "BUG000002",
    "itemId": "000002",
    "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1000",
    "itemLockId": 1719
  },
  "table": {
    "id": 1000,
    "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",
    "name": "Issues",
    "dbName": "UBG_ISSUES"
  },
  "fieldMetadata": [
    {
      "id": 75,
      "name": "Functional Area",
      "dbname": "FUNCTIONAL_AREA",
      "syscode": 0,
      "type": "selection",
      "required": false,
      "readonly": false,
      "renderHtml": false,
      "sectionId": 2,
      "helpText": "",
      "isMultiSelect": false,
      "isSearchable": false,
      "selections": [
        {
          "id": 0,
          "name": "(None)"
        },
        {
          "id": 27,
          "uuid": "18f578a6-4953-4008-9c96-bdae21e615cb",
          "name": "Administrator"
        },
        {
          "id": 28,
          "uuid": "a95673f2-5375-4eac-8070-4b4cd7391e7f",
          "name": "Help"
        },
        {
          "id": 29,
          "uuid": "fbb29892-a25e-4b4c-9fc2-54b89d697dc2",
```

```
        "name": "Setup"
      },
      {
        "id": 30,
        "uuid": "844cfc6d-64c3-45c4-94be-46dea304baf5",
        "name": "User Interface"
      },
      {
        "id": 64,
        "uuid": "ce247e8c-e9d6-44b4-8020-41c9e1ec9177",
        "name": "Compiled Modules"
      }
    ]
  },
  "fields": {
    "FUNCTIONAL_AREA": {
      "id": 0,
      "name": "(None)"
    }
  }
},
"result": {
  "type": "OK",
  "msg": ""
}
}
```

## Comments

None.

## SubmitToAux

Submits a new item into a specified auxiliary table.

## Parameters

Name	Type	Description
submittoaux	Function	Submits a new item into an auxiliary table. Call StartSubmitToAux first to get an item's field names or IDs.



table	<ul style="list-style-type: none"> <li>• ID (integer)</li> <li>• UUID (string)</li> <li>• internalname (string)</li> <li>• name (string)</li> </ul>	The table identifier.
-------	---	-----------------------

## Post Data

Name	Type	Description
itemOptions (optional)	<a href="#">ItemOptions [page 37]</a>	Use <code>itemOptions</code> to limit the response data.  When <code>itemOptions</code> are included, only the specified fields, plus fields required by the transition, are sent in the response.
transition	<a href="#">Transition [page 39]</a>	Use <code>transition</code> to provide new field values. Only provide values for fields that you want to change. Note that you must provide values for any required fields in the transition you are calling.

## Example

URL:

`http://AEServerName/jsonapi/submittoaux/UBG_IDM_CONFIGURATIONS`

Post data:

```
{fixedFields: false, transition: {"TITLE": "This is the title", "FIELD_NAME": "My Field"}}
```

Response:

```
{
  "item": {
    "id": {
      "id": 2,
      "uuid": "6deb6fdc-aa6d-4844-a78a-4e3b8ccea11be",
      "itemIdPrefixed": "",
      "itemId": "",
      "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1004"
    },
    "table": {
      "id": 1004,
      "uuid": "dcfead12-5fa4-471d-af37-7f9a36dee2c3",
      "name": "IDM Configurations",

```

```
"dbName": "UBG_IDM_CONFIGURATIONS"
},
"fieldMetadata": [
  {
    "id": 152,
    "name": "Application",
    "dbName": "TITLE",
    "syscode": 4,
    "type": "text",
    "renderHtml": false,
    "password": false,
    "maxLength": 80
  },
  {
    "id": 153,
    "name": "Field Name",
    "dbName": "FIELD_NAME",
    "syscode": 0,
    "type": "text",
    "renderHtml": false,
    "password": false,
    "maxLength": 80
  },
  {
    "id": 154,
    "name": "Clear Value",
    "dbName": "CLEAR_VALUE",
    "syscode": 0,
    "type": "text",
    "renderHtml": false,
    "password": false,
    "maxLength": 255
  },
  {
    "id": 155,
    "name": "Hidden Value",
    "dbName": "HIDDEN_VALUE",
    "syscode": 0,
    "type": "text",
    "renderHtml": false,
    "password": true,
    "maxLength": 127
  }
],
"fields": {
  "TITLE": {
    "value": "This is the title"
  },
  "FIELD_NAME": {
    "value": "My Field"
  },
  "CLEAR_VALUE": {
    "value": ""
  },
  "HIDDEN_VALUE": {
```

```
        "value": ""
    }
}
},
"result": {
    "type": "OK",
    "msg": ""
}
}
```

## Comments

None.

## SubmitToProject

Submits a new item into a specified project.

### Parameters

Name	Type	Description
submittoProject	Function	Submits a new item into a project. Call StartSubmitToProject first to get an item's field names or IDs.
project	<ul style="list-style-type: none"><li>ID (integer)</li><li>UUID (string)</li><li>internalname (string)</li></ul>	The project identifier.

### Post Data

Name	Type	Description
itemOptions (optional)	<a href="#">ItemOptions</a> <a href="#">[page 37]</a>	Use <code>itemOptions</code> to limit the response data. When <code>itemOptions</code> are included, only the specified fields, plus fields required by the transition, are sent in the response.
transition	<a href="#">Transition</a> <a href="#">[page 39]</a>	Use <code>transition</code> to provide new field values. Only provide values for fields that you want to change. Note that you must provide values for any required fields in the transition you are calling.

## Example

### URL:

```
http://AEServerName/jsonapi/submittoproject/UBG_ISSUES.ANIMATION_PRO
```

### Post data:

```
{transition:{"TITLE":"Browser hangs"}, fixedFields: false, fields: [{dbname:"TITLE"}]}
```

### Response:

```
{
  "item": {
    "id": {
      "id": 142,
      "uuid": "e3e94173-92e8-4c98-bd07-33277f58dd62",
      "itemIdPrefixed": "000196",
      "itemId": "000196",
      "url": "http://AEServerName:80/tmtrack/tmtrack.dll?...Template=view&TableId=1000"
    },
    "table": {
      "id": 1000,
      "uuid": "dc8cd329-b430-436f-bb75-bf90008e6a50",
      "name": "Issues",
      "dbName": "UBG_ISSUES"
    },
    "fieldMetadata": [
      {
        "id": 54,
        "name": "Title",
        "dbname": "TITLE",
        "syscode": 4,
        "type": "text",
        "renderHtml": false,
        "password": false,
        "maxLength": 80
      }
    ],
    "fields": {
      "TITLE": {
        "value": "Browser hangs"
      }
    }
  },
  "result": {
    "type": "OK",
    "msg": ""
  }
}
```

## Comments

None.

---

# JSON API Options

Use the following options when working with items and transitions.

- [ItemOptions](#) [page 37]
- [Transition](#) [page 39]

## ItemOptions

Limits the data that is returned with an item.

### Parameters

Name	Type	Description
<code>fields</code>	<ul style="list-style-type: none"><li>• ID (integer)</li><li>• UUID (string)</li><li>• dbname (string)</li><li>• name (string)</li></ul>	An array of objects used to specify the fields to display. Empty by default.
<code>fixedFields</code>	boolean	Combines with the <code>fields</code> array to determine the outputted fields. True by default. See usage notes below.
<code>includeNotes</code>	boolean	When true, the item's notes are included in the response. False by default.  The user's note preferences are honored, including the sort order and the number of notes to display. If the user has selected None as the display option for notes, no notes are returned.

### Usage

The `fixedFields` parameter combines with the `fields` array to determine the outputted fields as follows:

- `fixedFields = true` and `fields` is empty:  
System fields listed below are included in output.
- `fixedFields = true` and `fields` is populated:

System fields listed below and specified fields are included.

- `fixedFields = false` and `fields` is populated:  
Specified fields are included.
- `fixedFields = false` and `fields` is empty:  
All fields are included in output.
- `fixedFields` is unspecified and `fields` is populated:  
Specified fields are included.
- `fixedFields` is unspecified and `fields` is empty:  
System fields are included.

When `fixedFields = true`, the following system fields are included:

- Issue Type
- Project
- Title
- Description
- Submitter
- Last Modifier
- Submit Date
- Last Modified Date
- Active/Inactive
- State
- Owner
- Secondary Owner

## Example

Return all notes on the item, but only returns the "Title" and "Functional Area" fields:

```
{fixedFields: false, includeNotes: true, fields: [{id:"54"}, {id: "75"}]}
```

Response:

```
"fieldMetadata": [2]
0: {
  "id": 54
  "name": "Title"
  "dbname": "TITLE"
  "syscode": 4
```

---

```
"type": "text"
"renderHtml": false
"password": false
"maxLength": 80
}
1: {
  "id": 75
  "name": "Functional Area"
  "dbname": "FUNCTIONAL_AREA"
  "syscode": 0
  "type": "selection"
  "renderHtml": false
  "isMultiSelect": false
}
"fields": {
  "TITLE": {
    "value": "Image Builder crashes my system."
  }
  "FUNCTIONAL_AREA": {
    "id": 30
    "name": "User Interface"
  }
}
"notes": [1]
0: {
  "title": "Please update this ticket"
  "contents": "This ticket needs more details."
  "author": {
    "id": 8
    "name": "Administrator"
  }
  "modificationDateTime": {
    "svalue": "06/28/2016 02:48:23 PM"
    "value": 1467146903
  }
}
```

## Comments

- Because the `TS_ID` is subject to change when a process app is promoted, you may need to update the `fields` parameter with the `TS_ID` from the target environment after the promote is finished.
- If you specify the database name in `fields`, do not include the `TS_` prefix.
- Avoid using the field name when specifying each field to return in `fields`; the field name can be changed in SBM Composer at any time.

## Transition

Defines values for fields that you want to change in a transition

## Values

Name	Type	Description
E-signature values	<ul style="list-style-type: none"> <li>• <code>authUser</code> (string)</li> <li>• <code>authPassword</code> (string)</li> </ul>	Provide these values if the transition requires an electronic signature. For details on electronic signatures, refer to the <i>SBM Composer Guide</i> .
Fields	<ul style="list-style-type: none"> <li>• <i>Text, Numeric</i> (string)</li> <li>• <i>Binary/Trinary, Project, Single-Relational, Single Selection, User</i> (integer)</li> <li>• <i>Numeric</i> (integer or double)</li> <li>• <i>Multi-Selection, Multi-Relational, Multi-Group, Multi-User</i> (array of integers)</li> </ul>	Specify fields by the database field name (without the TS_ prefix).

## Usage

The `transition` object defines which fields you want to change as part of a transition.

When specifying values for each field, note the following:

- In Journal or append-only Journal fields, the string value is appended to the current value.
- In *Binary/Trinary* and selection fields, provide the value of the new selection. Refer to the StartTransition response data for available selections.
- In *Numeric* fields, send a integer for integer fields and a double for floating point or fixed precision fields. For string values, the new value will be parsed from the string.

## Example

The following example provides values for the Status Log and Tester fields:

```
{transition: {"STATUS_LOG": "Reopen issue", "TESTER": "7"}}
```

Response:



---

```
"fieldMetadata": [
  {
    "id": 65,
    "name": "Status Log",
    "dbname": "STATUS_LOG",
    "syscode": 0,
    "type": "journal",
    "renderHtml": false,
    "appendOnly": true,
    "sortNewestFirst": false
  },
  {
    "id": 63,
    "name": "Tester",
    "dbname": "TESTER",
    "syscode": 0,
    "type": "user",
    "renderHtml": false,
    "isMultiSelect": false
  }
],
"fields": {
  "STATUS_LOG": {
    "value": "07/13/2016 03:49:48 PM - Bill Admin:\r\n\r\nReopen issue\r\n"
  },
  "TESTER": {
    "id": 3,
    "name": "Chris Tester"
  }
}
```

## Comments

Integer values for field selections will be different in different environments. When you deploy or promote a process app, UUIDs are kept in sync but not TS\_IDs.