

Document number	Revision
DOCU12515	2

Transaction API

[TOC]

Introduction

The scope of this document covers the functionality of programmatically interact with Highstage items, specifically updating and creation.

The structure of the [Requests](#Request body definition) and the [Responses](#) are described in the following sections, as well as a [Powershell](#) code-snippet which uses them.

API Endpoint

The endpoint used for both updating and creation is the same, and can be found on your instance of Highstage under /ts/ts/ws/transaction.aspx (eg. https://highstage.company.com/ts/ts/ws/transaction.aspx) and requires a POST http request.

Request body definition

Name	Type	Description						
transaction	Element	Root element for each request						
object	Element	Child element of the transaction, one per item that requires update						
type	Attribute	The type attribute of object element specifies the object basetype						
		The command attribute of the object element specifies the operation to be done.						
command	Attribute	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>create</td> <td>Create a new object</td> </tr> <tr> <td>update</td> <td>Update an existing object</td> </tr> </tbody> </table>	Value	Description	create	Create a new object	update	Update an existing object
		Value	Description					
		create	Create a new object					
update	Update an existing object							
field	Element	Field element, the name* attribute defines the name of the field. The field content (InnerText) is the value of the field						

NOTE: Please note that the value of the name attribute is not necessarily the column header presented on the web interface, but the name of the column. See example below.

Reviewer	<input type="text"/>
Approver	<input type="text"/>

Reviewer

▼ **Details**

Title: Reviewer

Name: reviewby

Description:

OwnerType: DOCU

ObjType: ts_user

SqlType: varchar

SqlSize: 255

SqlLocation: [item].[reviewby]

QueryPath: reviewby

Xml: <column name="ReviewBy" title="Reviewer" description="" type="ts_u
objlist="1" sqltype="varchar" sqlsize="255" sqlnull="0" sqldefault="()"

Request samples

Request for creating a document

```
<?xml version="1.0" encoding="utf-8"?>
<transaction>
  <object type="doc" command="create">
    <field name="itemtype">NCR</field>
    <field name="workspace">QA</field>
    <field name="description">Non-conforming device performance measurement ref
61252</field>
    <field name="createby">TestUser</field>
  </object>
</transaction>
```

Request for updating a document's note and reviewer

```
<?xml version="1.0" encoding="utf-8"?>
<transaction>
  <object type="doc" id="GD14108-0A" command="update">
    <field name="note">failure to conform to accepted standards of
behavior</field>
```

```
<field name="reviewby">CHN</field>
</object>
</transaction>
```

Response

Name	Type	Description
response	Element	Root element for each response
severity	Attribute	Severity of the transaction result
		Value Description
		0 Normal
		1 Warning
		2 Error
3 Fatal Error		
xhtml	Element	Xhtml formatted message based on the severity returned

Response samples

Response with no errors

```
<response severity="0"></response>
```

Response with errors

```
<response severity="2">
  <xhtml>
    <![CDATA[<br/>ExecuteNode exception: System.Exception: Unknown column: notex.
Column does not exist in type: doc
   at TS.Object.Updater.Update(WebServiceResponse webServiceResponse,
SqlConnection connection, XmlNode objectNode) in
E:\TurboStage\TS\Object\Updater.cs:line 68
   at TS.Object.Transaction.Execute(XmlNode xmlNode) in
E:\TurboStage\TS\Object\Transaction.cs:line 172
]]>
  </xhtml>
</response>
```

Authentication

By default, Highstage has Windows authentication as the primary authentication mechanism, but usually there is also Basic authentication enabled for cross-platform support. Depending on the http client used, the implementation of the authentication might differ. The following section shows how to create an API request using the PowerShell as a scripting language.

PowerShell

```
$credentials=Get-Credential
$apiEndpoint='https://highstage.company.com/ts/ts/ws/transaction.aspx'
$body='<?xml version="1.0" encoding="utf-8"?><transaction><object type="doc"
command="update" id="GD11944-1A"><field name="workspace">TEST</field></object>
</transaction>'
$response=Invoke-WebRequest $apiEndpoint -Method POST -Credential $credential -
Body $body -UseBasicParsing | Select Content
```

Reading data

In order to fetch information from Highstage, two formats are supported XML and JSON each with small variations when it comes to the way the data is structured. The endpoint returning information can be found under /ts/ts/search.aspx (eg. <https://highstage.company.com/ts/ts/search.aspx>).

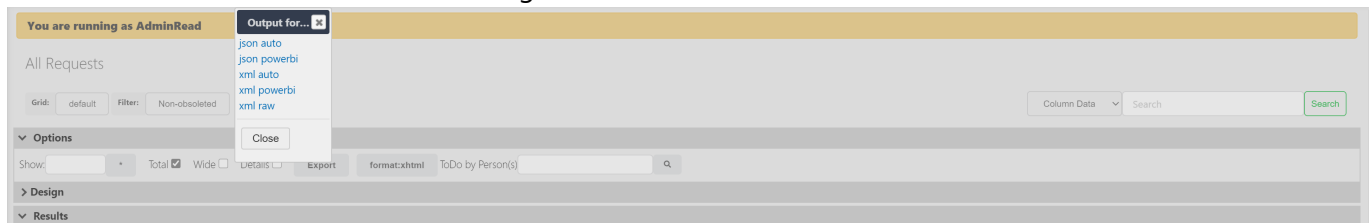
Request definition

Building the URL for a request definition can be done directly from the Highstage interface by searching and including appropriate filters and options required by the integration. Please refer to [*Searching in Highstage*] (https://www.turbostage.net/documentation/ts/search.aspx?&_grid=documentation&t=doc&_max=%2A&_title=Searching%20in%20Highstage&item=DOCU12184)

([https://www.turbostage.net/documentation/ts/search.aspx?&_grid=documentation&t=doc&_max=*%&_title=Searching in Highstage&item=DOCU12184](https://www.turbostage.net/documentation/ts/search.aspx?&_grid=documentation&t=doc&_max=*%&_title=Searching%20in%20Highstage&item=DOCU12184)) for a full introduction to searching for content in Highstage.

([https://www.turbostage.net/documentation/ts/search.aspx?&_grid=documentation&t=doc&_max=*%&_title=Searching in Highstage&item=DOCU12184](https://www.turbostage.net/documentation/ts/search.aspx?&_grid=documentation&t=doc&_max=*%&_title=Searching%20in%20Highstage&item=DOCU12184)) for a full introduction to searching for content in Highstage.

When requesting information the only mandatory filter is the type of the items requested (document, part, workspace etc), any other column filter being optional. After building the request, the output format should be chose, as it can be seen in the following :



The following formats can be used:

- XML output use - xml_auto (&_format= xml_auto)
- JSON output use - json_auto - (&_format=json_auto)

For Microsoft Power BI

Special formats have been developed for use with Microsoft Power Bi

- XML output use - xml_pbi (&_format= xml_pbi)
 - JSON output use - json_pbi - (&_format=json_pbi)
-



Highstage