

RESTful Web Service Interface

This topic explains how to import data into Futuro using Futuro's RESTful Web Service API.

The RESTful Web Service provides a way for an external system, such as a manufacturing execution system, to send data to Futuro. This interface can also be used to import data from JSON or other file formats into Futuro.

To authenticate the request to send the import data to Futuro, the RESTful Web Service can be configured to use **BASIC**, **TOKEN_BASED**, or **IDP_TOKEN** authentication. With BASIC authentication, the external system provides a Login Name, Password, and API Key to authenticate every request it sends to Futuro. With TOKEN_BASED authentication, the external system must first provide a Login Name, Password, and API Key to log into Futuro. Futuro will then issue a token that the external system must include when it sends the import data to Futuro. With IDP_TOKEN authentication, the external system must provide an authenticated JWT session token issued by a third-party Identity Provider (IDP) when it sends the import data to Futuro.

The RESTful Web Service can also be configured to process **Synchronous** or **Asynchronous** imports. For Synchronous imports, the data is validated and processed all the way into the appropriate Futuro tables and a confirmation is sent back to the external system. For Asynchronous imports, the data is placed in the Interface In Queue, and is not validated or processed immediately into the appropriate Futuro tables. For example, a manufacturing execution system sends Action records to Futuro and needs to know when a clock in or work order event was actually posted. In this case, a Synchronous import is used. However, when sending person records to Futuro, the external system only needs to know the data was received but not when it was processed. In this case, an Asynchronous import is used.

Configuration Steps:

[Configure the Server Settings](#)

[Configure the Certificate and Keystore for IDP_TOKEN Authentication](#)

[Configure the Interface Host](#)

[Configure the Import Definition](#)

[Configure the Distribution Model](#)

[Create and Send the Request](#)

[Configure and Run an Import Service \(Asynchronous\)](#)

See Also:

[Server Settings](#)

[Interface Host](#)

[Import Definition](#)

[Distribution Model](#)

[Service Instances](#)

[Enterprise Integration Manager](#)

Configure the Server Settings for RESTFUL_WEB_SERVICES

There are five [Server Settings](#) for the RESTful Web Service interface: `enabled`, `authentication_method`, `https`, `token_session_timeout_in_minutes`, and `basic_timestamp_offset_in_seconds`. You need to configure these settings as explained below.

1. Set enabled to TRUE

Set the **enabled** Setting Type to TRUE for the RESTFUL_WEB_SERVICES application.

2. Select the authentication method

The [authentication_method](#) Setting Type determines how the external system will be required to verify requests it sends to Futuro with the import data.

Set the **authentication_method** Setting Type to BASIC, TOKEN_BASED, or IDP_TOKEN for the RESTFUL_WEB_SERVICES application.

3. Configure the https Setting

Set the **https** Setting Type to TRUE or FALSE for the RESTFUL_WEB_SERVICES application.

If you set the [https Setting Type](#) to TRUE, then the header of the request that Futuro sends to the external system, and the payload data that the external system sends to Futuro, will be encrypted.

4. TOKEN BASED Authentication – Set the token session timeout in minutes

If you selected TOKEN_BASED as the authentication_method, you can change the [token session timeout in minutes](#). This setting defines how long a token will be valid once Futuro issues it. The default setting is 60 minutes.

Set the **token_session_timeout_in_minutes** for the RESTFUL_WEB_SERVICES application.

5. BASIC Authentication – Set the basic timestamp offset in seconds

If you selected BASIC as the authentication_method, you can set the [basic timestamp offset in seconds](#) Setting Type to a value other than DISABLED. When this setting is enabled, Futuro validates that the timestamp of the request from the external system is within this offset period from a timestamp configured in the request's header.

Set the **basic_timestamp_offset_in_seconds** to 5, 10, 30, 60, 120, or 300 seconds for the RESTFUL_WEB_SERVICES application.

Configure the Certificate and Keystore for IDP_TOKEN Authentication

If you are using IDP_TOKEN as your [authentication_method](#), you will need to obtain a certificate from your Identity Provider (IDP) and import it into a keystore.

When you save the keystore, take note of its password. You will need to add this to your Interface Host configuration.

The keystore file needs to be placed in the \certificate folder in your [FUTURO_HOME](#) folder.

The name of the keystore file and the name of the certificate alias will be used later in your Interface Host configuration.

Configure the Interface Host for the RESTful Web Service

Futuro includes an [Interface Host](#) for the RESTful Web Service imports called RWS_SERVER.

Make a **copy** of the *RWS_SERVER* Interface Host and save it with a new name.

Make sure the **Connection Type** is set to *Rest* and the **Host Type** is *Receiver*.

The configurations for the Interface Host will depend on your RESTFUL_WEB_SERVICES authentication_method Server Setting – [BASIC](#), [TOKEN_BASED](#), or [IDP_TOKEN](#).

BASIC Authentication

For BASIC authentication, enter the Interface Host's **Login Name** and **Password** that will be used to authenticate requests sent to Futuro.

In the [Host Parameters tab](#), add the GENERATE_API_KEY parameter. This parameter will create an API Key which you will include as a Header in your request.

You can also add the SYNCHRONIZATION_TYPE and LOAD_TYPE Host Parameters.

TOKEN_BASED Authentication

For TOKEN_BASED authentication, enter the Interface Host's **Login Name** and **Password** that will be used to authenticate requests sent to Futuro.

In the [Host Parameters tab](#), add the GENERATE_API_KEY parameter. This parameter will create an API Key which you will include as a Header in your request for the token.

You can also add the SYNCHRONIZATION_TYPE and LOAD_TYPE Host Parameters.

IDP_TOKEN Authentication

For IDP_TOKEN authentication, use the **Password** field in the Interface Host record to specify the keystore certificate password.

In the [Host Parameters tab](#), add the KEYSTORE_ALIAS and KEYSTORE_NAME parameters. See [Configure the Certificate and Keystore for IDP_TOKEN Authentication](#) for information on obtaining these values.

You can also add the SYNCHRONIZATION_TYPE and LOAD_TYPE Host Parameters.

Configure the Import Definition

When you send data to Futuro using a RESTful Web request, the Futuro [Import Definition](#) is used to process the data into the correct Futuro table.

The RESTful Web Service interface can be used with any [Import Type](#) (CSV, FIXED, JSON, WS, WSOAGIS, or XML).

The [Source Ind](#) for the Import Definition must be set to **In Queue**.

For XML, JSON, WS, and WSOAGIS imports, you can check the [Use Source Alias](#) box when configuring the [Source Fields](#) for the Import Definition. The Use Source Alias setting will generate a FreeMarker expression that will be used to find the Source Alias in the inbound record.

The following is an example of an Action import. The Import Type is JSON.

The records are being imported from a Shop Floor Execution system that is sending real-time labor transactions to Futuro. The Record Value Map is used to configure the GID and DID for each imported record.

The screenshot displays the 'Import Definition' configuration page. At the top, there are tabs for 'Find', 'Validate', 'Schedule Immediately', and 'Import File'. Below these, the 'Import Type' is set to 'JSON' and the 'Import Name' is 'RWS_SFE'. There is a 'Browse...' button for the 'Import File Name'. The 'Destination Records' section is active, showing a table with columns: Import Name, Context Name, Record Name, Sequence Num, XML Source Select, Skip Record, and Skip Empty. A single record is listed with 'RWS_SFE' as the Import Name, 'Action' as the Context Name, 'ACTION' as the Record Name, '0' as the Sequence Num, and 'action' as the XML Source Select. Below this is the 'Record Value Map' section, which has a table with columns: Import Name, Record Name, Field Name, Field Value Type, and Field Value. Two records are listed: one for 'RWS_SFE' with Record Name 'ACTION' and Field Name 'Did' having a 'STATIC' Field Value Type and value '8000'; the other for 'RWS_SFE' with Record Name 'ACTION' and Field Name 'Gid' having a 'STATIC' Field Value Type and value '8000'.

Import Name	Context Name	Record Name	Sequence Num	XML Source Select	Skip Record	Skip Empty
RWS_SFE	Action	ACTION	0	action	<input type="checkbox"/>	<input type="checkbox"/>

Import Name	Record Name	Field Name	Field Value Type	Field Value
RWS_SFE	ACTION	Did	STATIC	8000
RWS_SFE	ACTION	Gid	STATIC	8000

The Post Event Name field uses Field Translation records to map the single-character event names from the Shop Floor Execution system to the Futuro event names.

Destination Records

Source Fields

Query Fields

Lookup Fields

Find

Add

Modify

Delete

Download

Move Up

Move Down

<div><div></div></div>	Import Name	Mapped Fields	Formats	Source Alias	Source Rule
<div><div></div><div></div></div>	RWS_SFE	ACTION.Person Num		personNum	<#if (action.personNum)?has_content >\${a
<div><div></div><div></div></div>	RWS_SFE	ACTION.Post Event Name		eventName	<#if (action.eventName)?has_content >\${a
<div><div></div><div></div></div>	RWS_SFE	ACTION.Stage		eventStage	<#if (action.eventStage)?has_content >\${a

Field Map

Field Format

Field Translation

Find

Add




















Modify

Delete

Download

<div><div></div></div>	Source Value	Destination Value
<div><div></div><div></div></div>	B	BREAK
<div><div></div><div></div></div>	L	Work Order With Moves
<div><div></div><div></div></div>	M	Filler Time
<div><div></div><div></div></div>	U	MEAL

The Labor Type field also uses Field Translation records, in this case to map the three-character codes from the Shop Floor Execution system with the Futuro labor types.

Destination Records		Source Fields		Query Fields		Lookup Fields	
 Find	 Add	 Modify	 Delete	 Download	 Move Up	 Move Down	
	Import Name	Mapped Fields		Formats	Source Alias	Source Rule	
	RWS_SFE	ACTION.Stage			eventStage	<#if (action.eventStage)?has_content	
	RWS_SFE	ACTION.F Labor Type			laborType	<#if (action.laborType)?has_content >	
Field Map		Field Format		Field Translation			
 Find	 Add	 Modify	 Delete	 Download			
	Source Value	Destination Value					
	REG	Run					
	REW	Rework					
	SET	Setup					

Configure the Distribution Model for the RESTful Web Service

You will need to configure a [Distribution Model](#) record for the RESTful Web Service. The Distribution Model record contains the Receiver Name that is configured in the Interface Host. The Distribution Model also specifies the Interface Name and Transaction Name. It may include the Import Definition as well.

In the Distribution Model form, select the record with the Interface Name and Transaction Name that correspond to your import. Make a copy of the record. In this example, the Import Person record is being copied.

Copy Interface Distribution Model

Interface Name * Import

Transaction Name * Person

Interface Name Alias

Trans Name Alias

Sender Name * RWS_CLIENT

Receiver Name * RWS_IDP_TOKEN

Max Batch Trans * 1000

Batch Group By

Import Name Person_REST_Import

Export Context

Export Name

Logical Name person_rest_import_idp_token

Sequence Num 0

Enabled * ☒

Save Save and Add Cancel

In the **Receiver Name** field, select the [Interface Host record you configured earlier](#).

In the **Logical Name** field, enter a name to identify this record in the request. The Logical Name you define here will be included in the request URL.

You can select the **Import Name** to be used for the import. You can also specify an import name in the request URL. If you are sending records synchronously, you will need to specify the import name in the Distribution Record or in the request URL.

Make sure the record is **Enabled** and click Save.

Create and Send the Request

The configuration and steps for sending the request will depend on whether you are using BASIC, TOKEN_BASED, or IDP_TOKEN authentication.

Futuro's RESTful Web Service will accept POST and GET requests.

The screenshots below show requests being sent from an application called Postman. If you are using a different application to configure and send requests, your screens will be different.

Note: The RESTful Web Service API does not support importing datasets (a set of data from an external source that needs to be processed as one block within Futuro). For example, if the external system needs to perform a full sync of person records, all the data required to process the request as a full sync must be provided as part of a single request. Or, the first request from the external system has to identified as a full sync using the load_type parameter and all subsequent requests would need to be identified as incremental sync.

BASIC Authentication

This section explains how to configure a POST request to send data to Futuro using BASIC authentication.

Make sure the [authentication_method](#) is set to BASIC in your Server Settings.

Request URL

The format for the request URL is shown below. The URL will begin with *https* if HTTPS is enabled for the Restful Web Service application.

`http://<application server>:<port number>/IHTTPLCE/REST/import?logical_name=<logical name from Distribution Model>`

Example:

`https://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_basic`

The **logical_name** identifies the [Distribution Model record](#) that is used for the import. The Distribution Model record identifies the Interface Host you configured earlier. The Distribution Model also specifies the Interface Name and Transaction Name. It may include the Import Definition as well.

The request URL may also contain the following parameters:

import_name: For synchronous imports, you can include the name of the [Import Definition](#) that will be used to insert the data in the appropriate Futuro tables. This parameter will override the Import Name in the Distribution Model.

load_type: This parameter can be used to indicate whether the data being sent in the request will be processed as a Full Sync or Incremental Sync. This parameter will override the [LOAD_TYPE](#) configured in the Interface Host Parameter. Set the load_type parameter to full_sync or incremental_sync.

show_success_details: When a request is sent successfully, the response does not normally contain any details. You can set this parameter to true if you want to view details in the success response.

Example with additional parameters:

`https://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_basic&import_name=Person_REST_Import&load_type=full_sync&show_success_details=true`

Authorization and Headers

The **Authorization** Type for the request needs to be set to *Basic Authorization*. In the Username and Password fields, enter a Login Name and Password for an active Futuro Person record.

The screenshot shows the Futuro RESTful Web Interface. At the top, the title bar reads "RESTful Web Interface / BASIC person import". Below this, the request method is set to "POST" and the URL is "http://localhost:8080/IHTPLCE/REST/import?logical_name=person_rest_import_basic&show_success_details=true". A "Send" button is visible on the right. Below the URL bar, there are tabs for "Params", "Authorization", "Headers (13)", "Body", "Pre-request Script", "Tests", "Settings", and "Cookies". The "Authorization" tab is currently selected. It shows a "Type" dropdown set to "Basic Auth". Below this, a text box explains: "The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)". To the right, there are input fields for "Username" (containing "ADMIN") and "Password" (containing "5679").

The **Headers** shown below will also need to be configured for the POST request.

Content-Type

The Content-Type Header indicates the format of the data being sent in the request. The Futuro RESTful Web Service will only accept data with the same format as the Content-Type Header. Set the Content-Type to text/plain, application/xml, or application/json.

Api-key

The API Key is used to authenticate the request. The API Key comes from the [GENERATED API KEY parameter for the Interface Host](#).

Accept

If the Content-Type Header is set to text/plain, then the Accept Header needs to be set to application/xml or application/json. The Accept Header determines how the response from Futuro will be formatted. You can set the Accept Header to application/xml (to format the response in XML) or application/json (to format the response in JSON).

Timestamp

The Timestamp Header is only needed if you are using the [basic_timestamp_offset_in_seconds](#) setting. This setting causes Futuro to validate that the request's timestamp is within an offset period from the Timestamp specified in the request's Header.

If you are using the `basic_timestamp_offset_in_seconds` setting, enter the Timestamp in the following format: yyyy-MM-dd"THH:mm:ss"Z". For example: 2021-07-01T19:41:54Z represents July 1, 2021 07:41:54 PM UTC (03:41:54 PM EST).

AutoTime RESTful Web Interface / BASIC person import

POST http://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_basic

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies

Hide auto-generated headers

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Basic QURNSU46NTY3OQ==				
<input checked="" type="checkbox"/>	Postman-Token	<calculated when request is sent>				
<input checked="" type="checkbox"/>	Content-Type	text/plain				
<input checked="" type="checkbox"/>	Content-Length	<calculated when request is sent>				
<input checked="" type="checkbox"/>	Host	<calculated when request is sent>				
<input checked="" type="checkbox"/>	User-Agent	PostmanRuntime/7.32.2				
<input checked="" type="checkbox"/>	Accept	*				
<input checked="" type="checkbox"/>	Accept-Encoding	gzip, deflate, br				
<input checked="" type="checkbox"/>	Connection	keep-alive				
<input checked="" type="checkbox"/>	Content-Type	text/plain				
<input checked="" type="checkbox"/>	Api-key	caf7d704-4d58-453a-b50c-260d4626f...				
<input checked="" type="checkbox"/>	Accept	application/xml				

Request Body – Import Data

In the Body of the request, enter the records you are importing to Futuro. Make sure the format of the text matches the Content-Type you specified in the request Header.

AutoTime RESTful Web Interface / BASIC person import

POST http://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_basic&show_success_details=true

Params Authorization Headers (13) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Text

```

1 100,Alan,Shepard
2 101,Gus,Grissom
3 102,John,Glenn
4 103,Scott,Carpenter
5 104,Wally,Schirra
6 105,Gordon,Cooper
7 106,John,Young
8 107,James,McDivitt
9 108,Edward,White

```

Send the Request

When you send the request successfully, Futuro sends back a response indicating if the data was received in the Interface In Queue or if it was processed by a specific interface.

For example, the following is an XML formatted response to an asynchronous import of a person record. The Request URL had show_success_details set to True. The queue ID shown in the response can be used to look up the record in the Interface In Queue. You can also enter this queue ID in an Import Status Request.

```
"successCount": 1,
"successDetail": [
  {
    "queueId": 10017,
    "recordStatus": "R",
    "message": "Record successfully created."
  }
]
```

The following is an XML formatted response to a synchronous import of a person record (the Interface Host had its SYNCHRONIZATION_TYPE parameter set to SYNCHRONOUS). The Request URL had show_success_details set to True. The record was processed using the Person Interface. The queue ID shown in the response can be used to look up the record in the In XML Queue.

```
"successCount": 1,
"successDetail": [
  {
    "queueId": 10052,
    "sequenceNum": 12098,
    "returnCode": 20001,
    "message": "In . (?INTERFACE_PERSON? Successful)"
  }
],
"errorCount": 0
```

TOKEN_BASED Authentication

This section explains how to configure and send requests with data to Futuro using TOKEN_BASED authentication. You will need to configure (1) a POST request to login and obtain the token, (2) a POST request to send the data, and (3) a GET request to log out.

Make sure the [authentication_method](#) is set to TOKEN_BASED in your Server Settings.

Request to Obtain Token

You will need to create a POST request to log into Futuro and Futuro will respond with a token. This token will be used in the POST request that contains the data being sent to Futuro.

To create the request for the token, use the following URL. The URL will begin with *https* if HTTPS is enabled for the Restful Web Service application.

http://<application server>:<port number>/HTTPLCE/REST/login

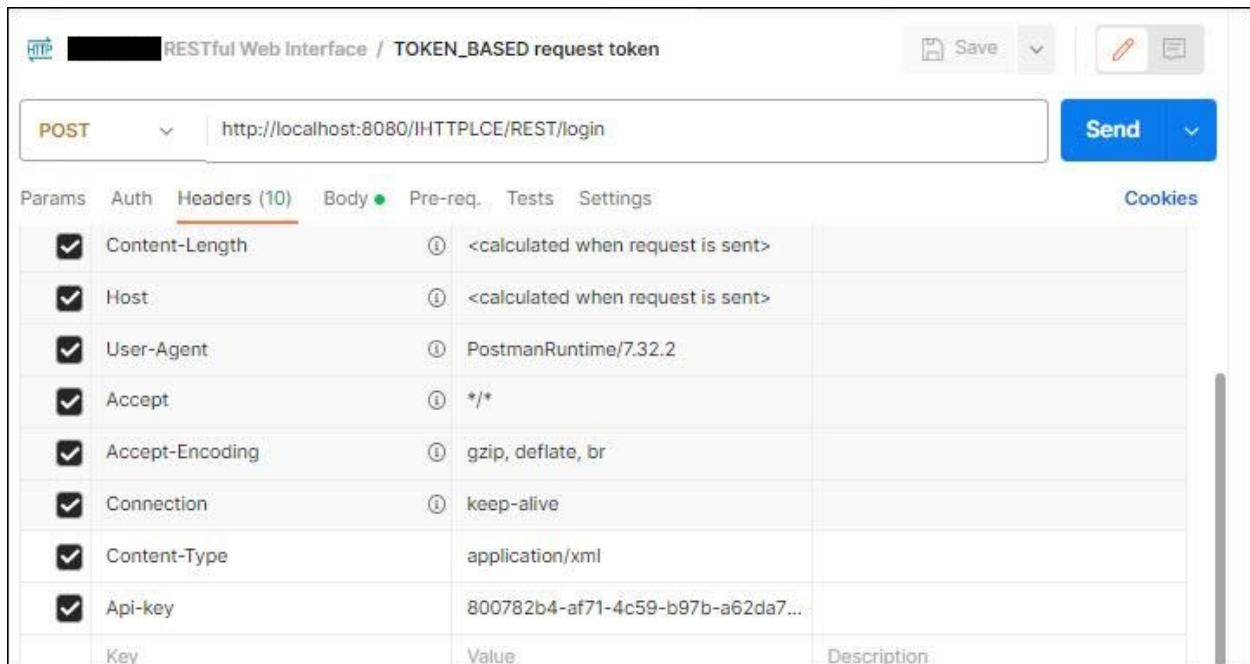
Example: <http://localhost:8080/IHTTPLCE/REST/login>

Login Request Headers

You need to configure the following Headers in your Login request:

Content-Type: The Content-Type Header indicates the format of the data being sent in the request. Set the Content-Type to application/xml, or application/json. The Content-Type should match the format of the data in the login request (see "Post Data" below).

Api-key: The API Key is used to authenticate the request. The API Key comes from the [GENERATED_API_KEY parameter for the Interface Host](#).



Post Data

The Body of the request must contain the information shown below. You can use the XML or JSON version. The username and password are the Login Name and Password from the [Interface Host you configured earlier](#).

XML format:

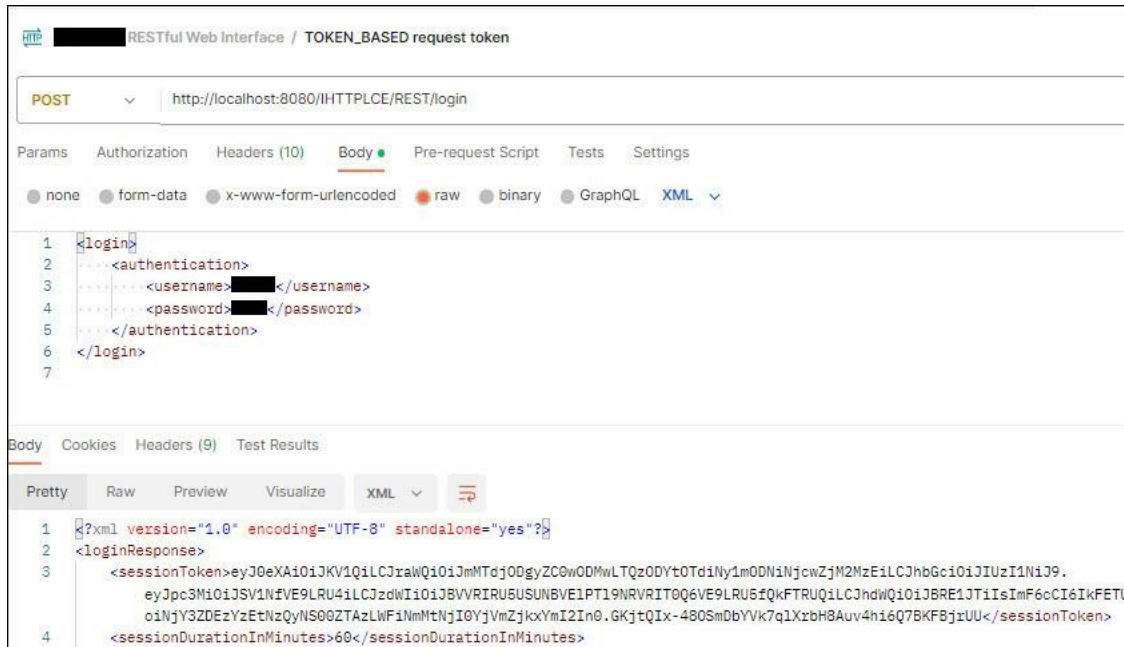
```
<login>
  <authentication>
    <username>futuro-user</username>
    <password>futuro-pwd</password>
  </authentication>
</login>
```

JSON format:

```
{
  "authentication": {
    "username": "futuro-user",
    "password": "futuro-pwd"
  }
}
```

```
}
}
```

When the request is sent successfully, the response will include a token. You need to copy this token so you can include it in the POST request's Authorization Header when you send the data to Futuro.



By default the token is valid for 60 minutes, unless you change the token_session_timeout_in_minutes Server Setting.

Request to Send Import Data

Once you obtain the token, you can create the POST request to send the data to Futuro.

Request URL

The format for the request URL is shown below. The URL will begin with *https* if HTTPS is enabled for the Restful Web Service application.

http://<application server>:<port number>/IHTTPLCE/REST/import?logical_name=<logical name from Distribution Model>

Example:

http://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_token

The **logical_name** identifies the [Distribution Model](#) record that is used for the import. The Distribution Model record identifies the Interface Host you configured earlier. The Distribution Model also specifies the Interface Name and Transaction Name. It may include the Import Definition as well.

The request URL may also contain the following parameters:

RESTful Web Interface / TOKEN_BASED person import

POST http://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_token

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

Header	Value	Description
<input checked="" type="checkbox"/> Content-Length	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.32.2	
<input checked="" type="checkbox"/> Accept	application/xml	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> Content-Type	text/plain	
<input checked="" type="checkbox"/> Accept	application/xml	
Key	Value	Description

Request Body – Import Data

In the Body of the request, enter the records you are importing to Futuro. Make sure the format of the text matches the Content-Type you specified in the request Header.

RESTful Web Interface / TOKEN_BASED person import

POST http://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_token

Params Authorization Headers (11) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL Text

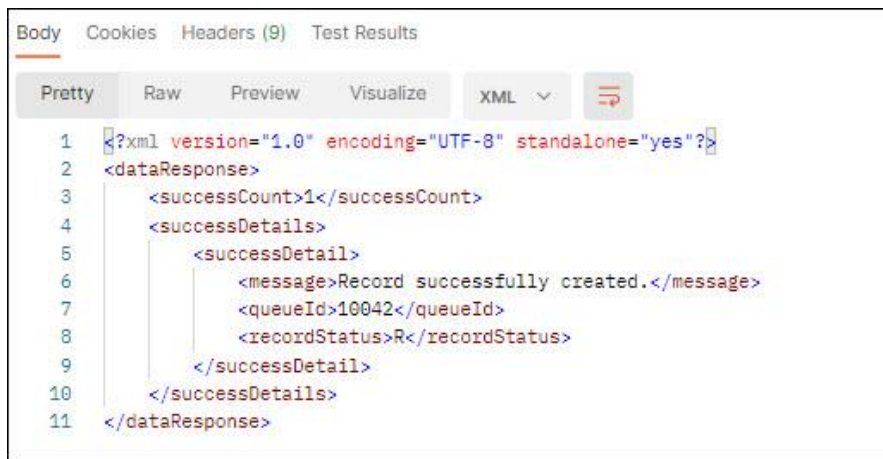
```

1 100,Alan,Shepard
2 101,Gus,Grissom
3 102,John,Glenn
4 103,Scott,Carpenter
5 104,Wally,Schirra
6 105,Gordon,Cooper
7 106,John,Young
8 107,James,McDivitt
9 108,Edward,White
10 109,Pete,Conrad
11 110,Frank,Borman
12 111,James,Lovell

```

Send the POST Request

When you send the request successfully, Futuro sends back a response indicating if the data was received in the Interface In Queue or if it was processed by a specific interface.



```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <dataResponse>
3   <successCount>1</successCount>
4   <successDetails>
5     <successDetail>
6       <message>Record successfully created.</message>
7       <queueId>10042</queueId>
8       <recordStatus>R</recordStatus>
9     </successDetail>
10  </successDetails>
11 </dataResponse>
```

Logout

When you are finished sending requests to Futuro with TOKEN_BASED authentication, you can send a GET request to log out of Futuro.

To create the Logout request, use the following URL. The URL will begin with *https* if HTTPS is enabled for the Restful Web Service application.

http://<application server>:<port number>/IHTTPLCE/REST/logout

Example:

https://qap-asgreh8:8543/IHTTPLCE/REST/logout

Logout Request Headers

You need to configure the following Headers in your Logout request:

Content-Type: Set the Content-Type to text/plain, application/xml, or application/json.

Authorization: Set the Authorization Type to Bearer Token and add the token from the previous request.

Accept: If the Content-Type Header is set to text/plain, then the Accept Header needs to be set to application/xml or application/json. The Accept Header determines how the response from Futuro will be formatted. You can set the Accept Header to application/xml (to format the response in XML) or application/json (to format the response in JSON).

IDP_TOKEN Authentication

This section explains how to configure and send requests with import data to Futuro using IDP_TOKEN authentication.

You will need to configure a POST request to obtain the token from the IDP and a POST request to send the data to Futuro.

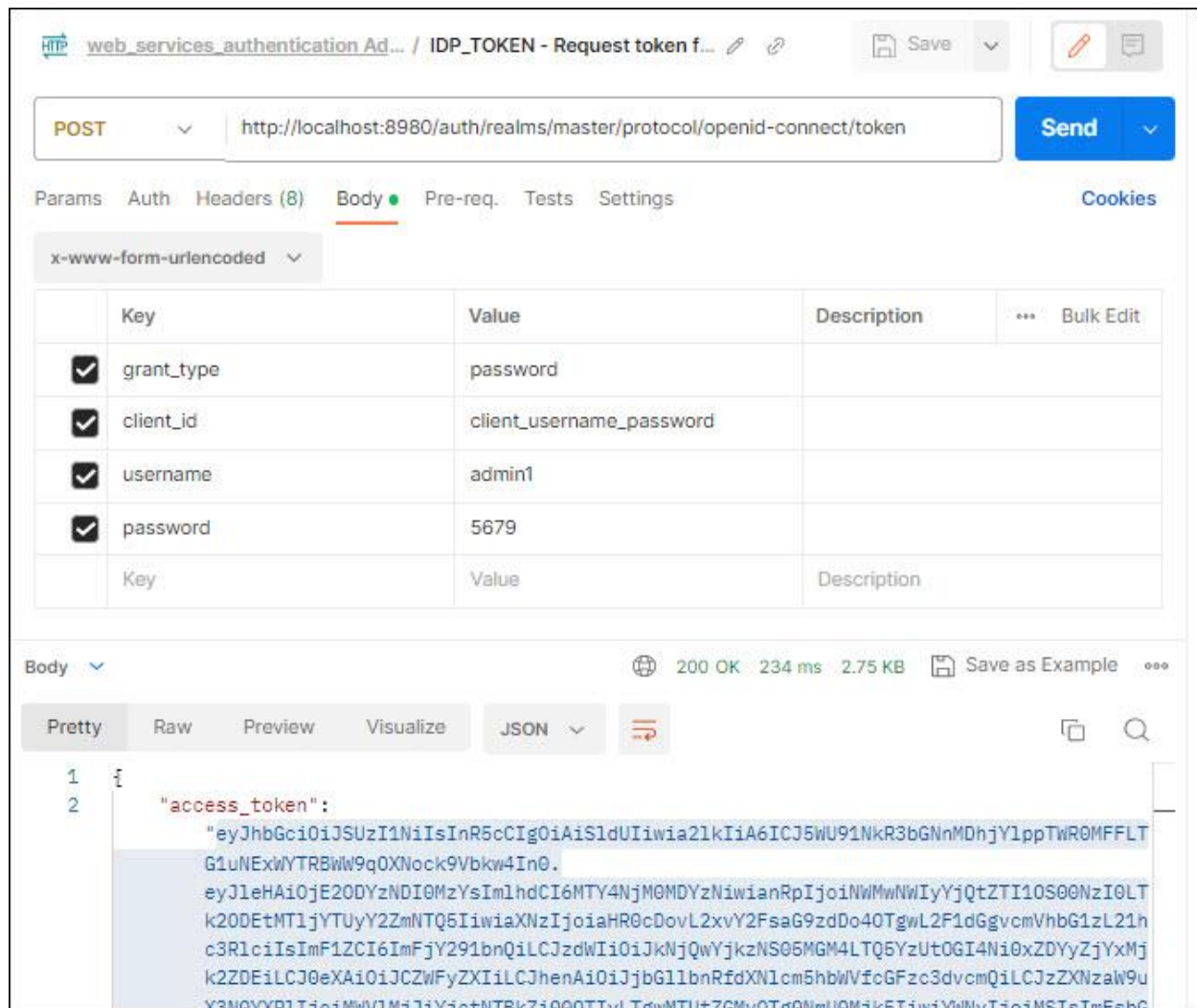
Make sure the [authentication_method](#) is set to IDP_TOKEN in your Server Settings; you have obtained a certificate from the IDP and imported it into a [keystore](#); and the [Interface Host](#) has been configured appropriately.

Obtain the Token from the IDP

You will need to obtain an authenticated JWT session token from your Identity Provider (IDP). This token must be included with every request you send to Futuro.

The method for obtaining this token will depend on which IDP you are using.

In the illustration below, a request is sent to a Red Hat SSO IDP and the token is returned. This request uses the Password configured in the Interface Host, as well as the KEYSTORE_ALIAS and KEYSTORE_NAME Interface Host Parameters, to obtain this token.



Create the Request URL

Once you obtain the token, you can create the POST request to send the data to Futuro.

The format for the request URL is shown below. The URL will begin with *https* if HTTPS is enabled for the RESTFUL WEB SERVICES application in your Server Settings.

Add another Header record for the **Accept** key. The Accept Header determines how the response from Futuro will be formatted. If the Content-Type Header is set to text/plain, then the Accept Header needs to be set to application/xml or application/json.

The screenshot shows the Postman RESTful Web Interface for a POST request to `http://localhost:8080/IHTTPLCE/REST/import?logical_name=person_rest_import_idp_token`. The **Headers** tab is selected, showing 11 headers. The **Accept** header is set to `application/json`.

Key	Value	Description
<input checked="" type="checkbox"/> Content-Type	text/plain	
<input checked="" type="checkbox"/> Content-Length	<calculated when request is sent>	
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.32.2	
<input checked="" type="checkbox"/> Accept	application/json	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> Content-Type	text/plain	
<input checked="" type="checkbox"/> Accept	application/json	

In the **Body** of the request, enter the records you are importing to Futuro. Make sure the format of the text matches the Content-Type you specified in the request Header. In the example below, the data is for a person import.

The screenshot shows the Postman RESTful Web Interface for the same POST request. The **Body** tab is selected, and the **raw** radio button is chosen. The body contains a list of 12 records, each on a new line, separated by a comma and space.

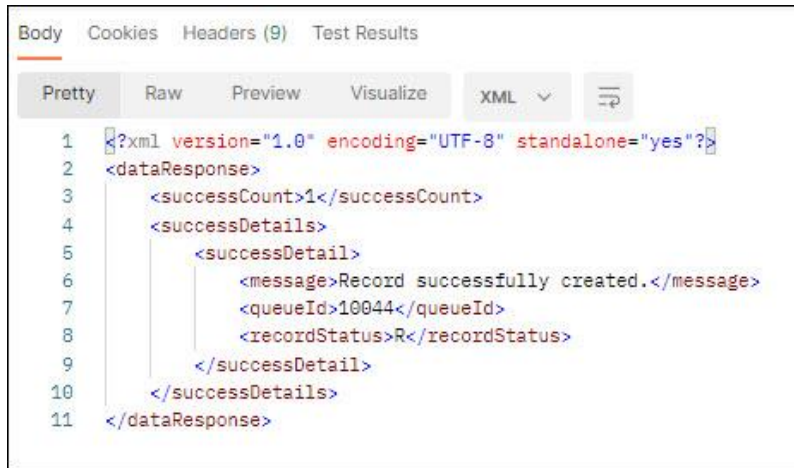
```

1 100,Alan,Shepard
2 101,Gus,Grissom
3 102,John,Glenn
4 103,Scott,Carpenter
5 104,Wally,Schirra
6 105,Gordon,Cooper
7 106,John,Young
8 107,James,McDivitt
9 108,Edward,White
10 109,Pete,Conrad
11 110,Frank,Borman
12 111,James,Lovell

```

Send the Request

When you send the request successfully, Futuro sends back a response indicating if the data was received in the Interface In Queue or if it was processed by a specific interface.



```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <dataResponse>
3   <successCount>1</successCount>
4   <successDetails>
5     <successDetail>
6       <message>Record successfully created.</message>
7       <queueId>10044</queueId>
8       <recordStatus>R</recordStatus>
9     </successDetail>
10  </successDetails>
11 </dataResponse>
```

Import Status Request

If you send a request asynchronously, you can use an Import Status Request to check the status of the record in the Interface In Queue. The Import Status Request is a GET type request.

To create the Import Status Request, use the following URL. The URL will begin with *https* instead of *http* if HTTPS is enabled for the Restful Web Service application.

http://<application server>:<port number>/IHTTPLCE/REST/import-status?queue_id=<queue id from the success message> &show_success_details=true

Example:

`https://localhost:8080/IHTTPLCE/REST/import-status?queue_id=10295&show_success_details=true`

If you are using TOKEN or IDP_TOKEN authentication, the **Authorization** section of your Import Status Request should have the same token you received from Futuro or your IDP.

You need to configure the following **Headers** in your Import Status request:

Content-Type: Set the Content-Type to text/plain, application/xml, or application/json.

Api-key: If you are using BASIC authentication, you need to include the Api-key Header in your Import Status request. The API Key is used to authenticate the request. The API Key comes from the GENERATED_API_KEY parameter for the Interface Host.

Configure and Run an Import Service (Asynchronous Imports)

If you are using the RESTful Web Service for asynchronous imports, the data will be placed in the Interface In Queue. You will need to run a service to move the data to the appropriate Futuro tables.

This section explains how to configure and run the IMPORT_FILES service to process the data in the Interface In Queue.

Make copy of the [IMPORT_FILES service](#) in the [Service Instance form](#).

Configure the service's parameters as follows:

- Set the LOAD_TYPE to Incremental Load or Full Load.
- Select the IMPORT_NAME that will be used to process the data into the appropriate Futuro table.
- Select the SENDER_NAME for this import. The Sender Name can be found on the Interface In Queue record.
- Set the TRANSACTION_GROUP to IMPORT.

When you run the service, the IN_CONVERT or IN_TEXT_CONVERT [task](#) will move the data from the Interface In Queue to the In XML Queue. Then the IN_XML_PROCESS task will move the data from the In XML Queue to the appropriate Futuro table based on the field mappings in the Import Definition.