

Konnect Rest API

- Introduction
 - Introduction
- Konnect Rest API
 - Konnect Rest API Sign Up
 - Test Connection to Konnect Rest Server
 - Authenticate Session for Rest Call
 - Make Transaction Rest Call
 - To push transaction data into Konnect ERP
 - Transaction - Production Confirmation of Components and FG using Serial No
 - Transaction - Sales Order/Purchase Order with Multiple Lines

Introduction

Introduction

Konnect Rest API Documentation

Version:- 1.0

Konnect ERP Provides REST API's for posting transactions from 3rd party systems via REST API, the following documentation describes the JSON Schema and URI's for the API's.

Please note that an active account in Konnect ERP system and credentials are mandatory for using this REST API.

Demo base URL:- <https://erpdemo.konnect-analytics.com>

Production base URL:- <https://erp.konnect-analytics.com>

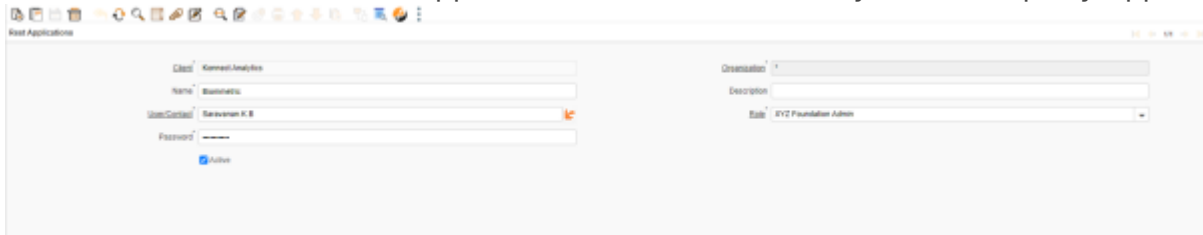
Konnect Rest API

Konnect Rest API Sign Up

Step 1:- How to Sign Up for Konnect Rest API?.

1) Login to the Konnect ERP System with the user having the required authorisation(typically user with Client-Admin included roles)

2) In Transaction Window Rest Applications, create a new entry for the 3rdparty Application.



3) Note that password and record_uu(appkey) as per screen shot below, these are the credentials to be used for Rest communication. Please note that the user id mapped in this window would be the User name used for recording all transactions using this Rest app.

KA_Rest_UU=dd7b39e1-4edc-4d5a-8618-dcd7e7622ecd



4) Using the AppKey and password Rest Calls can be made to Konnect ERP

Test Connection to Konnect Rest Server

Step 2:- Test Connection to Konnect Rest Server

- At Any time a connection test can be made to the Konnect Server without any credentials
- If the server is up and running a success message would be sent as a reply.

URI:- /api/app/check

Content-Type:- application/json

Method:- GET

Request Header:- None

Response Payload:-

Response Sample(Success):-

```
{  
  
  "id": "",  
  
  "error": "",  
  
  "msg": "Api version 1.0"  
}
```

Authenticate Session for Rest Call

Step 3:- Authenticate Session for Rest Call

1) For any Transaction API Call , you need to have a valid token obtained through this authentication API

2) The Timeout for this token is 10 minutes,if a rest api call has not been made for 10 mins the Token is invalidated on the server side and you have to request for a new one using this api

URI:- /api/app/authenticate

Content-Type:- application/json

Method:- GET

Request Header:-

Attribute	Description
appkey	The appkey obtained in Step 1(rest_uu)
password	The password maintained for this app in Rest applications window

Response Payload:-

Attribute	Description
id	Blank String
error	If there are errors in the appkey or password this field is populated, else it is blank

msg	If request is success, the login token is returned in this field
-----	--

Response Sample(Success):-

```
{  
  
  "id": "",  
  
  "error": "",  
  
  "msg": "e1TMMBDZUsCTOJVQJeiaKwW0"  
}
```


Make Transaction Rest Call

Step 4:- Make Transaction Rest Call

1) Once Rest Authentication is made, transaction calls can be made using the logintoken obtained from Step 3

2) If the server returns a 401 error with message Token is Invalid, then re-authentication needs to be done to get a new login token

3) For all transactions under the same URI, you can use GET to get a sample payload to use for posting data using POST method.

URI:- /api/app/system/* (detail URI explained later for each transaction)

Content-Type:- application/json

Method:- GET (to get a sample payload)

Request Header:-

Attribute	Description
appkey	The appkey obtained in Step 1(rest_uu)
logintoken	The token obtained in Step 3

Response Payload:- (as per transaction described later in this section)

Attribute	Description
id	ID of transaction is request is success
error	If there are errors in the transaction this field is populated, else it is blank

msg	If request is success, this will contain transaction messages
data	This part would vary for each transaction, depending on the transaction

Response Sample(Success):-

```
{
  "id": "",
  "error": "",
  "msg": "",
  "sernos": {
    "FG Serno1": [
      "Comp1 Serno",
      "Comp2 Key/Search key",
      "Comp3 Serno"
    ],
    "FG Serno2": [
      "Comp4 Key/Search Key",
      "Comp5 Serno",
      "Comp6 Serno"
    ]
  }
}
```


To push transaction data into Konnect ERP

Method: POST (to push transaction data into Konnect ERP)

Request Header:-

Same as GET method

Request Payload:-

As per Response Payload for same URI in GET Method

Response Payload:-

Response Payload :-

Same as response but with the ID/MSG and error populated as per the scenario.

Transaction - Production Confirmation of Components and FG using Serial No

Transactions :-

The transactions section explains the transactions that can be posted via Rest and the payload details and other vital information required by 3rd party systems to integrate into Konnect ERP.

Production Confirmation of Components and FG using Serial No.:-

URI:- /api/app/system/prodconfirm

Sample Payload:-

```
{  
  
  "id": "",  
  
  "error": "",  
  
  "msg": "",  
  
  "sernos": {  
  
    "FG Serno1": [  

```

"Comp1 Serno",

"Comp2 Key/Search key",

"Comp3 Serno"

],

"FG Serno2": [

"Comp4 Key/Search Key",

"Comp5 Serno",

"Comp6 Serno"

]

}

}

Payload Attribute	Type (JAVA)	Remarks	Validations
sernos	Map<String,ArrayList<String>>	Map of each FG Serno with the list of Component Serial Nos or Component Search Keys	All Values should be Strings
<FG Serno>	Place holder for the FG Serial Number	Should be a valid Serial Number	The serial Number should be mapped to atleast 1 Manufacturing Order in ERP
<Comp Serno>	Place holder for the Component Serial Number or Component Product key(if serial number does not exists)	Should be a valid Serial number or Product Key	The Component should be part of the Bom in the manufacturing order of the parent

Transaction - Sales Order/Purchase Order with Multiple Lines

Sales Order/Purchase Order with Multiple Lines:-

This transaction is used to post data for Sales order or Purchase Order with all the relevant information from 3rd party systems

URI:- /api/app/system/salesorder

Sample Payload:-

```
{  
  
  "id": "",  
  
  "error": "",  
  
  "msg": "",  
  
  "documentno": "",  
  
  "bpartner": "Customer Key/Search Key",  
  
  "ordertype": "Document Type Name",  
  
  "bpartner_location": "BP Order Address Name",
```


"orderdate": 1696271400000,

"promiseddate": 1696271400000,

"pricelist": "Price List Name(if empty default is used)",

"warehouse": "Name of Warehouse",

"paymentterm": "Payment Term Name(default immediate)",

"description": "Order Description",

"ordersource": "Order Source Name",

"salestransaction": true,

"lines": [

{

"lineno": 10,

"description": "Line Description",

"productkey": "Product Key/Search Key",

"chargename": "Charge Name(when Product is blank)",

"promiseddate": 1696271400000,

"taxname": "Tax Category Name",

"qty": 0.0,

"price": 0.0

},

```

{

    "lineno": 20,

    "description": "Line Description2",

    "productkey": "Product Key/Search Key",

    "chargename": "Charge Name(when Product is blank)",

    "promiseddate": 1696271400000,

    "taxname": "Tax Category Name",

    "qty": 0.0,

    "price": 0.0

}

]

}

```

Payload Attribute	Type (JAVA)	Remarks	Validations
documentno	String	If Order Posted successfully the document no would be returned here	
bpartner	String	Customer/Vendor Search Key	The value is validated against existing Values
ordertype	String	Document Type Name of Sales Order or Purchase Order	The value is validated against existing Values
bpartner_location	String	Business partner location name, if left blank it would be taken from default values	Optional

orderdate	Long(Timestamp value)	Date of Order	
promiseddata	Long(Timestamp value)	Required Date of Order	
pricelist	String	Price List Name(if empty default is used)	Optional
warehouse	String	Name of Warehouse	
paymentterm	String	Payment Term Name(default immediate)	Optional
description	String	Order Description	Optional
ordersource	String	Order Source Name	Optional
salestransaction	Boolean	True for Sales order/False for Purchase order	
lines	ArrayList<Lines)	Contains arraylist of the lines of the order	Atleast 1 line required to create the order
lineno	Integer	Sequence of the Line	
description	String	Line Description	Optional
productkey	String	Product Key/Search Key	Product or Charge is mandatory
chargename	String	Charge Name(when Product is blank)	Product or Charge is mandatory
promiseddate	Long(TimeStamp)	Promised Date of Line	
taxname	String	Tax Category Name	Optional, else will be taken from product master
qty	Double	Qty of the Product	
price	Double	Price of Item	Amount before tax is calculated as qty*price

