CENTRALSQUARE
TECHNOLOGIES

Finance Enterprise Web Services

Release 19.1

Contact Us

CentralSquare Technologies
1000 Business Center Drive
Lake Mary, Florida 32746
centralsquare.com

Documentation

This document is maintained by the Documentation team. To contact the Documentation team, email documentation@centralsquare.com.

Connect Community

To access additional documentation, articles, and forums, visit community.centralsquare.com.

Education and Training

To contact the Customer Education team, email csu@centralsquare.com or call (800) 727-8088.

To find continuing education opportunities, visit train.superion.com/csu.

Support

To access the Support Portal, visit support.centralsquare.com.

To contact the Support team, call 800-851-4800.

Legal notices

Table of Contents

Overview

This document outlines the process of calling web services using the Central Square Technologies Finance Enterprise suite. The client assembly details that provide access to all of the web service methods are included. There is a ClientProxyFactory class that assists with building the client proxies used for connecting to the web services. Also included is the management interface to the web services, the security and connection information, and the details of the request context. Finally, a detailed list of the supported web services is included with the data contracts that define the data exchanged between a service and the client. The Web Service location can be found at:

```
http://<hostname>/Finance/services/WcfProxy/CentralSquareWcfWebProxy.svc
```

Client Assembly

CentralSquare Technologies provides an assembly that gives access to all of the finance web services. The assembly file, **… WcfClientProxy.dll**, is located in the **Finance/bin** directory on the finance web server. Including this assembly as a reference to a .NET project allows access to the various web service methods.

Note: You also need the following additional references:

- .NET System.Runtime.Serialization
- System.ServiceModel components

ClientProxyFactory

ClientProxyFactory is a static factory class in the …WcfClientProxy.dll assembly. The purpose of this factory is to provide client proxies to each of the groups of web services. You should configure two properties before using the client proxy factory.

1. TargetServer - The TargetServer points to the finance server.
2. Timeout property - The Timeout property is the default number of milliseconds before a timeout occurs. If not set, the timeout defaults to thirty seconds.

For each of the groups of web services (service contracts) there is a method on the ClientProxyFactory class named **Create<ContractName>Client** where the contract name is the name of the service contract. Once the client proxy has been generated for the specific service contract, you can use it to call directly in to the web services. After you finish using the client proxy, you should close it.

Management Interface

The management interface provides the ability to download the latest version of the client assembly as an array of bytes. This can be useful during development to ensure you are using the most recent client assembly from the finance server.

Security and Connection Information

Before calling any of the web services, the user must be identified, and the connection configured. To perform this function, use the Authorize service contract **CreateAuthorizeClient** with one of the following methods.

1. ConnectionList() - This method provides a list of connection names that are available.
2. SimpleValidate (ref string authtoken, string usersid, string connection) - This method validates that the user is able to connect and provides an authentication token that can be used to make future requests. The **usersid** is the Security Identifier (SID) of the connecting user, and the **connection** is the name of the connection. The method returns true if the user is valid.

Otherwise, the method returns false.

3. Validate (string usersid, string connection) - Much like SimpleValidate, this method validates that the user is able to connect. However, this method also returns some additional information about the user, such as the default GL and JL ledger codes, the default timeout (in seconds), and the authentication token.

Once the user has been validated, the authentication token can be included in the request context and requests can be made.

Request Context

Each request, with the exception of the management interface, requires a request context to be included. The request context is a standard way to keep track of errors that occur during the processing of a request. Because each of the requests potentially contains output information, a new request context should be used with each request.

The request context includes the following fields:

- AuthToken (Input) - This property contains the authentication token required to make a request, as provided from the SimpleValidate and Validate methods in the CreateAuthorizeClient service.
- ContainsErrors (Output) - If errors exist as a result of processing the request, this property is returned true; otherwise it is false.
- CurrentLedgers (Input) - In situations where the request may utilize default ledger codes, this property can provide those values. Unless there is a specific reason not to, CentralSquare Technologies recommends that this property be the same value as derived from the default ledgers when using the “Validate” method to authorize a user.

- Events (Output) – This property returns an array of warning, error, or informational messages that occurred during the processing of the request.
- IgnoreWarnings (Input) - The **IgnoreWarnings** property can be set to true or false. If the ignore warnings flag is set to false, then any warnings which can occur during the processing of the request causes the activity to fail. Setting **IgnoreWarnings** to true allows processing to continue, even if warnings do exists. Regardless, any errors always result in the activity failing, and the appropriate messages arereturned in the **Events** property. If the **IgnoreWarnings** property is left blank, the default behavior is as if it were set to false.
- Mask (Input) - This property is not currently used by any web services. It is possible, though not likely, that a business rule could use this value when creating new records. In the future this might be used at the web service level to indicate authorization to call a specific web service method by a specific user.
- SubSystem (Input) - This property is not currently used by any web services. It is possible, though not likely, that a business rule could use this value when creating new records.
- Timeout (Input) - The number of milliseconds to wait before assuming that the request is not returning. Unless a different default value is specified in the ClientProxyFactory, the default value is “30000”, which equates to 30 seconds.
- TraceEnabled (Input) - Currently for internal use only.
- TraceLevel (Input) - Currently for internal use only.

An extension method is also included for the request context to authenticate. By calling the authenticate method on a request context, you can provide a user SID and a connection name, and the request context is populated for you using the default settings for the user after an authentication is made. Each call to the Authenticate method returns a new copy of a request context. However, the authenticate call to the web service is only made once per user. CentralSquare Technologies recommends that you use the “Authenticate” method from the request context to build the request methods.

Examples

Example #1

This first example demonstrates the **KeyLookup** method in the

AccountLookupServiceClient, using a matching operator.

> **Note**: In this simple example a specific user SID is passed along with the connection (prod), instead of deriving the SID based on the user.

```
// Example of calling the KeyLookup method in the AccountLookupServiceClient
// web service to return Finance Org Keys that match the value “101*”
using System;
using System.Security.Principal;
using ....PS.Wcf.Client;
using ....PS.WcfCommon;
using  WcfGL;

class Program

{
    public static void Main()
    {
        // Create the client proxy to connect to the web
        service. ClientProxyFactory.TargetServer = "<server
        name>"; AccountLookupServiceClient proxy =
            ClientProxyFactory.CreateAccountLookupServiceClient();

        // Build up the request context that contains the connection information.
        // In this example, just use a given SID and the “prod”
        connection. RequestContext requestContext = new RequestContext();
        requestContext.Authenticate(
            "S-1-5-21-123123123-123123123-123123123-
            1002", "prod");

        // Build the request to find all Org Keys that start with
        “101”. KeyLookupParameter param = new KeyLookupParameter();
        param.Field = KeyLookupParameterField.OrgKey;
        param.Operator = LookupOperator.Like;
        param.Value = "101*"; KeyLookupRecord[]
        result = proxy.KeyLookup(
            ref requestContext, 0, "GL",
            new KeyLookupParameter[] { param });

        // Verify that we have a result.
        if (!requestContext.ContainsErrors)
        {
            // Display the results; in this case the Org Keys and
            Descriptions.  Console.WriteLine("Key  Data:");
            foreach (KeyLookupRecord rec in result)
            {
                Console.WriteLine(“{0} - {1}”, rec.OrgKey, rec.OrgKeyLongDesc);
            }
        }

        // Always close the proxy when the request is
        complete. proxy.Close();
    }
}
```

Example #1 console window output:

```
 Key Data:
101200   - Payroll
101250   – Purchasing
101300   – Human Resources
101400   – Police Department
101500   – Fire Department
101600   – Main Street Library
101610   – Wilson Library
101611   – Library Control
```

Example #2

This second example outlines:

Using the client proxy factory

- Building the request context
- Executing the web service request that creates an AR/CR record in Finance
- Parsing the response from the request.

```
using System;
using System.Security.Principal;
using SunGard.PS.Wcf.Client;
using SunGard.PS.WcfCommon;
using WcfAR;

class Program
{
    static void Main(string[] args)
    {
        try
        {
            // Create the client proxy to connect to the web
            service. ClientProxyFactory.TargetServer = "<server
            name>"; ARCRServiceClient proxy =
                ClientProxyFactory.CreateARCRServiceClient();

            // Build the request context that contains the connection
            information. string connection = GetConnection();
            RequestContext requestContext =
                BuildRequestContext(connection);

            // Build the AR/CR record that will be created.
            ARCRRecord arcrRecord = BuildInsertRecord();

            // Run the insert.
            ARCRResponse result = proxy.InsertAndValidateArCrRecord( ref
                requestContext, arcrRecord);

            // Parse the response.
            ParseResponse(result, requestContext);

            // Always close the proxy when the request is
            complete. proxy.Close();
        }
```

```
       catch (Exception except)
       {
           // Report each of the exceptions in the exception
           stack. while (except != null)
           {
               Console.WriteLine(except.Message);
               Console.WriteLine();
               except = except.InnerException;
           }
       }
   }

   // Parse the response and spit out any errors.
   private static void ParseResponse(ARCRResponse result,

    RequestContext requestContext)
{
    // Check if there were errors.
    if (requestContext.ContainsErrors)
    {
        // Loop through each event/error.
        foreach (Event evnt in requestContext.Events)
        {
            // Display the type of event and event message.
            Console.WriteLine("{0}: {1}", evnt.Level, evnt.Message);
        }
    }

    // Verify that the response exists, and if so, display the
    results. if (result != null)
    {
        Console.WriteLine("Set Type: {0}", result.SetType);
        Console.WriteLine("  Set ID: {0}", result.SetID);
        Console.WriteLine(" Receipt: {0}", result.ReceiptNumber);
    }
    else
    {
        Console.WriteLine("The response was null.");
    }
}

// Construct the data for the new record.
private static ARCRRecord BuildInsertRecord()
{
    // Create the record instance.
    ARCRRecord record = new ARCRRecord();

    // Populate the record data.
    // Many of the fields in the resulting record will be defaulted
    in. record.SetType = "CR";
    record.GlGr = "GL"; record.GlKey
    = "101500"; record.GlObj =
    "6000"; record.TermCode = "T";
    record.UnitPrice = 15.25m;
    record.UserReceiptTotal =
    15.25m; record.HitAR = false;
```

```
     return record;
 }

 // Select the name of the connection to
 use. private static string GetConnection()
 {
     // Get a list of available connections.
     AuthorizeClient authClient =
         ClientProxyFactory.CreateAuthorizeClient();
     string[] connections =
     authClient.ConnectionList(); authClient.Close();

     // Return the first connection available.
     return connections[0];
 }

 // Build the request context object that will contain the connection info.
 private static RequestContext BuildRequestContext(
     string connection)
 {
        // Create the new instance.
        RequestContext requestContext = new RequestContext();
        // Get the user SID and authenticate.
        string sid = WindowsIdentity.GetCurrent().User.Value;
        requestContext.Authenticate(sid, connection);

        return requestContext;
    }
}
```

Example #2 console window output:

```
     Set Type:     CR
     Set ID:       WSCR4792
     Receipt:      CR750985
```

Notice that when building the record, the Set ID and Receipt number was not given. A business rule was in place to derive these values from a seed generator. Because of this, it is important to display this information back to the user, so they know what to look for in Finance Enterprise.

Available Web Services

The information displayed in this section is also available directly from the web server at the following location:

https://fusion.centralsquare.com

Account Lookups

PostAccountLookup

Use this method to lookup an account in the ledger.

Sample Code: C#

```
using System.Net;

string uri =
"http://localhost/FusionOSServices/v1/ONESolution/AccountLookup/Lookup";
var xmlText =
"<AccountLookupRequest><Ledger>GL</Ledger><Year>2015</Year><AccountLookupParame
ter><Field>OrgKey</Field><Operator>6</Operator><Value>101*</Value></AccountLook
upParameter></AccountLookupRequest>"
using (WebClient req = new WebClient())
{
    string stringResult = wc.UploadString(new Uri(uri), "POST", xmlText);
    XmlDocument response = xmlDoc.LoadXml(stringResult);
    // TODO
}
```

PostKeyLookup

Use this method to lookup a key in the ledger.

Sample Code: C#

```
using System.Net;

string uri =
"http://localhost/FusionOSServices/v1/ONESolution/AccountLookup/KeyLookup";
var xmlText =
"<KeyLookupRequest><Ledger>GL</Ledger><KeyLookupParameter><Field>OrgKey</Field>
<Operator>6</Operator><Value>101*</Value></KeyLookupParameter></KeyLookupReques
t>"
using (WebClient req = new WebClient())
{
    string stringResult = wc.UploadString(new Uri(uri), "POST", xmlText);
    XmlDocument response = xmlDoc.LoadXml(stringResult);
    // TODO
}
```

PostObjectLookup

Use this method to lookup an object in the ledger.

Sample Code: C#

```
using System.Net;

string uri =
"http://localhost/FusionOSServices/v1/ONESolution/AccountLookup/ObjectLookup";
var xmlText =
"<ObjectLookupRequest><Ledger>GL</Ledger><IncludeNonTransactional>true</Include
NonTransactional><MaxRows>10</MaxRows><ObjectLookupParameter><Field>ObjGroup1</
Field><Operator>6</Operator><Value>101*</Value></ObjectLookupParameter></Object
LookupRequest>"
using (WebClient req = new WebClient())
{
    string stringResult = wc.UploadString(new Uri(uri), "POST", xmlText);
    XmlDocument response = xmlDoc.LoadXml(stringResult);
    // TODO
}
```

PostGetAccounts

Use this method to lookup accounts in the ledger.

Sample Code: C#

```
using System.Net;

string uri =
"http://localhost/FusionOSServices/v1/ONESolution/AccountLookup/Accounts";
var xmlText =
"<AccountLookupRequest><Ledger>GL</Ledger><IncludeNonTransactional>true</Includ
eNonTransactional><MaxRows>10</MaxRows><AccountLookupParameter><Field>OrgKey</F
ield><Operator>6</Operator><Value>101*</Value></AccountLookupParameter></Accoun
tLookupRequest>"
using (WebClient req = new WebClient())
{
    string stringResult = wc.UploadString(new Uri(uri), "POST", xmlText);
    XmlDocument response = xmlDoc.LoadXml(stringResult);
    // TODO
}
```

PostGetKeys

Use this method to lookup keys in the ledger.

Sample Code: C#

```
using System.Net;

string uri =
"http://localhost/FusionOSServices/v1/ONESolution/AccountLookup/Keys";
var xmlText =
"<KeyLookupRequest><Ledger>GL</Ledger><IncludeNonTransactional>true</IncludeNon
```

```
Transactional><MaxRows>10</MaxRows><KeyLookupParameter><Field>OrgKey</Field><Op
erator>6</Operator><Value>101*</Value></KeyLookupParameter></KeyLookupRequest>"
using (WebClient req = new WebClient())
{
    string stringResult = wc.UploadString(new Uri(uri), "POST", xmlText);
    XmlDocument response = xmlDoc.LoadXml(stringResult);
    // TODO
}
```

PostGetObjects

Use this method to lookup objects in the ledger.

Sample Code: C#

```
using System.Net;

string uri =
"http://localhost/FusionOSServices/v1/ONESolution/AccountLookup/Objects";
var xmlText =
"<ObjectLookupRequest><Ledger>GL</Ledger><IncludeNonTransactional>true</Include
NonTransactional><MaxRows>10</MaxRows><ObjectLookupParameter><Field>ObjGroup1</
Field><Operator>6</Operator><Value>101*</Value></ObjectLookupParameter></Object
LookupRequest>"
using (WebClient req = new WebClient())
{
    string stringResult = wc.UploadString(new Uri(uri), "POST", xmlText);
    XmlDocument response = xmlDoc.LoadXml(stringResult);
    // TODO
}
```

Accounts Payable

InsertApRecords

WcfAP.APServices.InsertApRecord (WcfAP.ApRecord[])

This method inserts but does not validate an array of Accounts Payable records. Removing validation from the process allows for better performance when dealing with very large numbers of transactions. The primary purpose for this service is the creation of refunds but it also can update vendor information.
Parameters:

records:	(input) Array of AP records
returns:	Array of AP responses for each of the record inserts

InsertAndValidateApRecord

WcfAP.APServices.InsertAndValidateApRecord(WcfAP.ApRecord)

This method is identical to the previous service – InsertApRecords. This request processes a single record at a time, performing standard data validation prior to inserting the record.

Parameters:

record:	(input) AP record
returns:	AP response for the inserted record

Data Contract - WcfAP.ApRecord

Name	Type / Size	Description
BatchId	Char(16)	Set ID - Multiple records are grouped together in what is called a set or batch. If this field is left blank, the set ID is automatically generated. Note: We are currently transitioning from a maximum of 8 characters to 16 characters.
Desc	Char(30)	Description - Transaction description. This field cannot be larger than 30 characters.
Div	Char(4)	Division - AP transactions are assigned to a specific division. Division codes allow for separation and grouping of transactions for reporting purposes.
FinCd	Char(8)	Finance code - Finance codes act as a link or shortcut to a financial account number, such as the org key or object code. Including a pre-defined finance code allows the following six properties to be left blank – G1Gr, G1Key, G1Obj, J1Gr, J1Key, J1Obj.
GlGr	Char(2)	GL ledger code - General ledger code for this entry. If blank, the user’s default code is used. A finance code can also be used in this field.
GlKey	Char(10)	GL organization key - The general ledger organization key for this entry. A finance code can also be used in this field.
GlObj	Char(8)	GL object code - The general ledger object code for this entry. A finance code can also be used in this field.
JlGr	Char(2)	JL ledger - The job ledger code for this entry. If blank, the user’s default code is used. A finance code can also be used to default into this field.
JlKey	Char(10)	JL organization key - The job ledger organization key for this entry. A finance code can also be used to set this field.
JlObj	Char(8)	JL object code - The job ledger object code for this entry. A finance code can also be used in this field.
Wo	Char(12)	Work order - If applicable, a work order number can be passed to the journal entry transaction.
MiscCd	Char(4)	Misc code - A four-character miscellaneous code to track transactions as needed.
Ref	Char(16)	Invoice - Invoice number. If blank, AP assigns this number.

Name	Type / Size	Description
RefDate	dateTime	Invoice date - Transaction reference date. If blank, AP inserts the current date.
RefundType	Char(2)	Refund Type - Type of refund.
RelOne	Char(2)	Relate 1 - Relate code used for grouping transactions. These are heavily used in determining AP transactions that relate to 1099-MISC.
RelTwo	Char(2)	Relate 2 - Relate code used for grouping transactions. These are heavily used in determining AP transactions that relate to 1099-MISC.
SecRef	Char(116)	Second Ref - Secondary reference field.
TrnsFmt	Char(2)	Transaction Format - Transaction formats which describes how the transaction description is stored in the general ledger after being posted. The default value is **NB**.
Units	Num(20,5)	Quantity – Number of transactions. This field which defaults to **1**.
UnitPrice	Num(20,5)	Unit Price - Unit price of the transaction. This field is required.

Data Contract - WcfAP.PeSummary

Name	Type	Description
PeId	Char(12)	Customer - Vendor ID.
PeName	Char(50)	Name - Customer name.
PeAddrCd	Char(2)	Address code - Address code tied to the customer address.
PeAddr1	Char(50)	Address - Customer address line 1.
PeAddr2	Char(50)	Address - Customer address line 2.
PeAddr3	Char(50)	Address - Customer address line 3.
PeAddr4	Char(50)	Address - Customer address line 4.
PeCity	Char(30)	City - Customer city.
PeState	Char(2)	State - Customer state code.
PeZip	Char(20)	ZIP - Customer zip code
PedbCd	Char(1)	PEDB - Customer Pedb code. This should be set to **P**.

Accounts Receivable / Cash Receipts

InsertArCrRecords

WcfAR.ARCRService.InsertArCrRecords(WcfAR.ARCRRecord[])
This method inserts, but does not validate an array of Accounts Receivable / Cash Receipt records. Removing validation from the process allows for better performance when dealing with very large numbers of transactions.

Parameters

records:	(input) Array of AR/CR records
returns:	Array of AR/CR responses for each of the record inserts

InsertAndValidateArCrRecords

WcfAR.ARCRService.InsertAndValidateArCrRecord(WcfAR.ARCRRecord)
Identical to the previous service, this request processes a single record at a time, performing standard data validation prior to inserting the record.

Parameters

record:	(input) AR/CR record
returns:	AR/CR response for the inserted record

Data Contract - WcfAR.ARCRRecord

Name	Type / Size	Description
BatchId	Char(16)	Set ID - Multiple records are grouped together in what is called a set or batch. If blank this number is generated for you. **Note**: CentralSQuare is currently transitioning from a maximum of 8 characters to 16 characters.
TrType	Char(2)	Transaction Type - Whether the record came from AR or CR. **AR** and **CR** are the only two possible values for this property.
AcctId	Char(12)	Account ID – Account ID assigned to the customer this request is associated with. This is required for AR transactions but optional for CR transactions.
AcctName	Char(30)	Account name – Name of the customer the request is associated with. This field is optional for CR transactions.
FinCd	Char(8)	Finance code – Finance code acts as a link or shortcut to the financial account number, such as the org key or object code. Including a pre-defined finance code allows the following six properties to be left blank – G1Gr, G1Key, G1Obj, J1Gr, J1Key, J1Obj.
GlGr	Char(2)	GL ledger code - If left blank, the user's default code is used. A finance code can also be used to set this field.
GlKey	Char(10)	GL organization key - General ledger organization key for this entry. A finance code can also be used to set this field.
GlObj	Char(8)	GL object code - General ledger object code for this entry. A finance code also can be used to set this field.
JlGr	Char(2)	JL ledger - Job ledger code for this entry. If left blank, the user's default code is used. A finance code can also be used to default this field.
JlKey	Char(10)	JL organization key - Job ledger organization key for this entry. A finance code can also be used to set this field.
JlObj	Char(8)	JL object code - Job ledger object code for this entry. A finance code can be used to set this field.
Wo	Char(12)	Work order - Work order number can be passed into the journal entry transaction, if applicable.
Term	Char(8)	Term code - Associates the transaction to a specific term within the

Name	Type / Size	Description
		Accounts Receivable subsystem. Terms are most commonly associated with higher education clients. Most organizations use a single term. This field is only required for AR transactions.
Div	Char(4)	Division - AR transactions are assigned to a specific division. Division codes allow for separation and grouping of transactions for reporting purposes. This property is optional for CR transactions.
Ref	Char(16)	Invoice - Invoice number. This field is required for AR transactions but optional for CR transactions. When used for CR, it links a payment to a specific invoice.
Ref2	Char(16)	Receipt - Receipt number. This field required for CR transactions but should never be used for AR transactions.
Desc	Char(30)	Description - Transaction description. This field should be no larger than 30 characters.
RefDt	dateTime	Invoice or receipt date - If left blank, AR inserts the current date.
MiscCd	Char(4)	Misc code - Four-character miscellaneous code is available to track transactions as needed.
Qty	Num(20,5)	Quantity - Transaction quantity. If this field is blank it defaults to **1**.
UnitPrice	Num(20,5)	Unit Price - Unit price of the transaction. This field is required.
PayType	Char(2)	Payment type - Cash Receipts payment type, such as CA for Cash.
PayRef	Char(16)	Payment ref - Payment reference for CR transactions, such as check number **6030**.
BankId	Char(2)	Bank ID - Bank ID within Finance Enterprise.
BankSlip	Char(10)	Bank slip – Number that identifies a specific bank deposit. This field is only used for CR.
BankDt	dateTime	Bank date - Date that the deposit was made to the bank. This field is only used for CR.
AddrCd	Char(2)	Address code - Address code for the customer to identify the address which is to be associated with this transaction.
PedbCd	char(1)	PEDB - The PEDB code indicates the source of the customer. If this is a CentralSquare client, this field is **P**.
TrnsFmt	Char(2)	Trans format - Transaction formats that describe how the transaction description is stored in the General Ledger application after being posted to the General Ledger. The default value is **NB**.
HitAr	Char(1)	Hit AR – Whether this value should be stored in the Accounts Receivable database. Set this value to **Y** to store it in the database. This field should always be set to **Y** for AR transactions.

Delete ArCrRecords

WcfAR.ARCRService.DeleteArCrRecords(WcfAR.ARCRRecord[])
You use this method to delete an array of Accounts Receivable or Cash Receipts records.

Parameters

records:	(input) Array of AR/CR records
returns:	Array of AR/CR responses for each of the record deleted

GetFinanceARCRDetails Method

WcfAR.ARCRService.GetFinanceARCRDetailsMethod(WcfAR.ARCRRecord[])
You use this method to get all ARCR transactions for a given ARCR Account ID. The returning array contains the transactions ordered by invoice, type, receipt, or receipt date. The type is 'AR' or 'CR'. The 'AR' type records represent charges. 'CR' type records represent cash receipts.

Parameters

records:	(input) Account ID, InvoiceID. The account/Person ID is required. The Invoice ID is optional.
returns:	An array of transaction records for all activities found in the system for the Account/Person ID specified.

GetFinanceARCRPersonList Method

WcfAR.ARCRService.GetFinanceARCRPersonListMethod (WcfAR.ARCRPersonListRequest)
You use this method to get an array of account summary records that matches a given name. The names of existing accounts found in OS Finance will be searched based on the search parameters passed into the method.

Parameters

records:	(input) Account ID, Name, Ledger
returns:	A response which includes a summary of matching records for the search performed. Use the CustomerID field returned to perform further inquiries about AR account invoice summary details, or full transaction details.

GetReceipts Method

WcfAR.ARCRService.GetReceiptsMethod (WcfAR.ARCRReceiptListRequest)
Get all CR transactions based on lookupRequest. The returning array will contain the transactions order by invoice, type, receipt, and receipt date. The 'CR' type records represent cash receipts.

Parameters

records:	(input) Account ID, Receipt Number, Receipt Date, Receipt Text
returns:	An array of transactions records for all activity found in the system for the specified lookup parameters.

GetUnPaidCharges Method

WcfAR.ARCRService.GetUnPaidCharges Method(WcfAR.ARCR[])
Get an array of open charges associated with a particular AR/CR account and invoice.

Parameters

records:	(input) Account ID, Invoice ID
returns:	It returns an array of ARChargeItems.

BenTek Interface

BenTekEmployee

This service returns the following information about employees: employee ID, First Name, Last Name, email address department and calendar.

Name	Type	Description
	String	If an employee ID is provided only information for that employee will be returned.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri =
"http://localhost/FusionOSServices/v0/ONESolution/BenTek/Employees";


    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        while (uri != null)
        {
            string result = wc.DownloadString(uri);

            var response = JObject.Parse(result);

            var count = (string)response["RequestResponse"]["Count"];
            var employees =
(JContainer)response["RequestResponse"]["DATA"]["EMPLOYEES"];
```

```
            foreach (var employee in employees)
            {
                // Your code goes here
                string fName = (string)employee["FNAME"];
                string lName = (string)employee["LNAME"];
            }
            // responses with more then 1000 employees are broken up into 
multiple requests
            uri = (string)response["RequestResponse"]["NextUri"];
        }
    }

}
```

Sample Response: JSON

```
{
    "RequestResponse": {
        "Count": "1",
        "DATA": {
            "EMPLOYEES": [
                {
                    "CALENDAR": "STANDARD",
                    "DEPARTMENT": "03",
                    "EMAIL": "eramund@qmail.com",
                    "FNAME": "ENRIQUE",
                    "HDT": "5/21/2013 12:00:00 AM",
                    "IDA": "E01010",
                    "LNAME": "RAMUNDSEN"
                }
            ]
        }
    }

}
```

BenTekEmployeePCN

This service returns the following information about employees: employee ID, First Name, Last Name, email address department and calendar.

Name	Type	Description
	String	If an employee ID is provided only information for that employee will be returned.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri = 
"http://localhost/FusionOSServices/v0/ONESolution/BenTek/Employees/JOBS";
```

```
            using (WebClient wc = new WebClient())
            {
                wc.Headers.Add("Content-Type", "application/json");
                // Replace "ID" with supplied AppID
                wc.Headers.Set("X-APPID", "ID");
                // Replace "KEY" with supplied AppKey
                wc.Headers.Set("X-APPKEY", "KEY");

                while (uri != null)
                {
                    string result = wc.DownloadString(uri);

                    var response = JObject.Parse(result);

                    var count = (string)response["RequestResponse"]["Count"];
                    var employees =
        (JContainer)response["RequestResponse"]["DATA"]["EMPLOYEES"];

                    foreach (var employee in employees)
                    {
                        // Your code goes here
                        string employeeId = (string)employee["IDA"];

                        foreach(var job in employee["PCN"])
                        {
                            string jobDescription = (string)job["LONGDESC"];
                            string departement = (string)job["DEPARTMENT"];
                            string recordType = (string)job["RECTYPE"];
                            string supervisorId = (string)job["SUPERID"];
                            foreach(var hour in job["HOURS"])
                            {
                                string hourNumber = (string)hour["CDHNO"];
                                string hourDescription = (string)hour["CDHTITLE"];
                                string hourType = (string)hour["TCPTYPE"];
                            }
                        }
                    }
                    // responses with more then 1000 employees are broken up into
        multiple requests
                    uri = (string)response["RequestResponse"]["NextUri"];
                }
            }

        }
```

Sample Response: JSON

```
        {
            "RequestResponse": {
                "Count": "1",
                "DATA": {
                    "EMPLOYEES": [
                        {
                            "DEPARTMENT": "04",
                            "IDA": "E00011",
                            "PCN": [
{
    "DEPARTMENT": "",
    "IDA": "E00011",
    "PCN": "000041",
    "RECTYPE": "PM",
    "LONGDESC": "ACCOUNTING TECHNICIAN",
    "SUPERID": "E00001",
    "HOURS": [
        {
            "HOUR": [
                {
                    "CDHNO": "3002",
                    "CDHTITLE": "SALARY",
                    "TCPTYPE": "REG"
                },
                {
                    "CDHNO": "3200",
                    "CDHTITLE": "OVERTIME",
                    "TCPTYPE": "OT"
                },
                {
                    "CDHNO": "3270",
                    "CDHTITLE": "FLSA1 OVERTIME -MISC",
                    "TCPTYPE": "OT2"
                },
                {
                    "CDHNO": "3402",
                    "CDHTITLE": "VACATION USED",
                    "TCPTYPE": "LEAVE"
                },
                {
                    "CDHNO": "3422",
                    "CDHTITLE": "SICK USED",
                    "TCPTYPE": "LEAVE"
                },
                {
                    "CDHNO": "3432",
                    "CDHTITLE": "SICK USED-FAMILY",
                    "TCPTYPE": "LEAVE"
                },
                {
                    "CDHNO": "3442",
                    "CDHTITLE": "PERSONAL USED",
                    "TCPTYPE": "LEAVE"
                },
                {
                    "CDHNO": "3488",
                    "CDHTITLE": "COMP USED",
                    "TCPTYPE": "LEAVE"
                }
            ]
        }
    ]
}
```

```
                    ]
                }
            ]
        }
    }
}
```

BenTekEmployeeFamily

This service returns the following information about employee family: employee id,family member dob,family member first middle and last name,Address 1, Address 2,city,state,zip,relation,age,gender,phonecd, and status.

Name	Type	Description
	String	If an employee ID is provided only information for that employee will be returned.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri = "https://localhost:44398/v1/ONESolution/BenTek/Family/E01010";


    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        while (uri != null)
        {
            string result = wc.DownloadString(uri);

            var response = JObject.Parse(result);

            var count = (string)response["RequestResponse"]["Count"];
            var family =
(JContainer)response["RequestResponse"]["DATA"]["FAMILY"];

            foreach (var member in family)
            {
                string famId = (string)member["IDA"];

                // Your code goes here
            }
            // responses with more then 1000 family members are broken up into
multiple requests
```

```
                uri = (string)response["RequestResponse"]["NextUri"];
            }
        }

    }
```

Sample Response: JSON

```
    {
        "RequestResponse": {
            "Count": "6",
            "DATA": {
                "FAMILY": [
                    {
                        "IDA": "E01010",
                        "GENDER": "F",
                        "MISCSTAT": "N",
                        "PHONECD": "C1",
                        "RELATION": "SP",
                        "FNAME": "Rebecca",
                        "MIDDLENAME": "Jackie",
                        "LNAME": "De Marco",
                        "SUFFIX": ""
                    },
                    {
                        "IDA": "E01010",
                        "GENDER": "F",
                        "MISCSTAT": "N",
                        "PHONECD": "C1",
                        "RELATION": "PA",
                        "FNAME": "Helen",
                        "MIDDLENAME": "Elaine",
                        "LNAME": "Lawson",
                        "SUFFIX": ""
                    },
                    {
                        "IDA": "E01010",
                        "GENDER": "M",
                        "MISCSTAT": "",
                        "PHONECD": "",
                        "RELATION": "PA",
                        "FNAME": "Henry",
                        "MIDDLENAME": "",
                        "LNAME": "Lawson",
                        "SUFFIX": ""
                    },
                    {
                        "IDA": "E01010",
                        "GENDER": "M",
                        "MISCSTAT": "B",
                        "PHONECD": "C1",
                        "RELATION": "CH",
                        "FNAME": "Tyrion",
                        "MIDDLENAME": "Kid",
                        "LNAME": "Handfull",
                        "SUFFIX": ""
                    },
                    {
                        "IDA": "E01010",
                        "GENDER": "F",
                        "MISCSTAT": "",
                        "PHONECD": "C1",
                        "RELATION": "CH",
                        "FNAME": "Rreir",
                        "MIDDLENAME": "",
                        "LNAME": "Sox",
                        "SUFFIX": ""
                    },
                    {
                        "IDA": "E01010",
                        "GENDER": "F",
                        "MISCSTAT": "",
                        "PHONECD": "C1",
                        "RELATION": "CH",
                        "FNAME": "Lady",
                        "MIDDLENAME": "",
                        "LNAME": "Gordo",
                        "SUFFIX": ""
                    }
                ]
            }
        }

    }
```

GetEmployeeBenefits

This service returns the following information about employees: employee ID, benefet plan, benefet plan begin date and benefit plan end date.

Name	Type	Description
	String	If an employee ID is provided only information for that employee will be returned.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri =
"https://localhost:44398/v1/ONESolution/BenTek/Employees/Benefits/E01010";


    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        while (uri != null)
        {
            string result = wc.DownloadString(uri);

            var response = JObject.Parse(result);

            var count = (string)response["RequestResponse"]["Count"];
            var benefits = 
(JContainer)response["RequestResponse"]["DATA"]["HRBENEINFOS"];

            foreach (var benefit in benefits)
            {
                string empId = (string)member["IDA"];

                // Your code goes here
            }
            // responses with more then 1000 benefit records are broken up into 
multiple requests
            uri = (string)response["RequestResponse"]["NextUri"];
        }
    }

}
```

Sample Response: JSON

```
 {
     "RequestResponse": {
         "Count": "5",
         "DATA": {
             "HRBENEINFOS": [
                 {
                     "BENEBEG": "7/1/2009 12:00:00 AM",
                     "BENEEND": "12/31/2050 12:00:00 AM",
                     "BENEPLAN": "BLPDPT00",
                     "IDA": "E01010"
                 },
                 {
                     "BENEBEG": "12/14/2013 12:00:00 AM",
                     "BENEEND": "12/31/2050 12:00:00 AM",
                     "BENEPLAN": "SLSPATSP",
                     "IDA": "E01010"
                 },
                 {
                     "BENEBEG": "1/1/2016 12:00:00 AM",
                     "BENEEND": "12/22/2016 12:00:00 AM",
                     "BENEPLAN": "ADBPATE0",
                     "IDA": "E01010"
                 },
                 {
                     "BENEBEG": "12/23/2016 12:00:00 AM",
                     "BENEEND": "12/31/2050 12:00:00 AM",
                     "BENEPLAN": "ADBPATE0",
                     "IDA": "E01010"
                 },
                 {
                     "BENEBEG": "7/7/2017 12:00:00 AM",
                     "BENEEND": "12/31/2050 12:00:00 AM",
                     "BENEPLAN": "BNSWPTE0",
                     "IDA": "E01010"
                 }
             ]
         }
     }

 }
```

PutEmployeeBenefits

This service writes benefit assignment info to Finance Enterprise.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri = "https://localhost:44398/v1/ONESolution/BenTek/Benefits/";
    string data = "[" +
                  "  {" +
                  " \"EntityId\": \"ROOT\", " +
                  " \"Id\" : \"E01010\", " +
                  " \"ActionCd\" : \"NEW\", " +
                  " \"BeneBeg\": 07/07/2017\" +
                  " \"BeneEnd\" : \"12/31/2050\", " +
                  " \"BenePlan\" : \"BNSWPTE0\", " +
                  " \"ApprvCd01\" : \"SEND\", " +
                  "  } " +
                  " , " +
                  "  { " +
                  " \"EntityId\": \"ROOT\", " +
                  " \"Id\" : \"E01011\", " +
                  " \"ActionCd\" : \"NEW\", " +
                  " \"BeneBeg\": 06/01/2017\" +
                  " \"BeneEnd\" : \"12/31/2050\", " +
                  " \"BenePlan\" : \"DENTPTE0\", " +
                  " \"ApprvCd01\" : \"WAIT\", " +
                  "  } " +
                  "  ]";

    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        string result = wc.UploadString(uri, "PUT", data);

        var response = JObject.Parse(result);
        var count = (string)response["RequestResponse"]["Count"];

        // verify count is expected number
        }
    }
}
```

Sample Response: JSON

```
{
    "RequestResponse": {
        "Count": "2",
        "DATA": null
    }

}
```

Document Online

AttachDocumentMethod

WcfDO.DocumentOnlineService.AttachDocumentMethod
You use this method to attach the specified document to record.
Parameters
records: (input) DocID, history key, record, and instance key

DeleteDocumentMethod

WcfDO.DocumentOnlineService.DeleteDocumentMethod
You use this method to delete the specified document attached to the record.
Parameters
records: (input) DocID, history key, and instance key

GetAttachDefinition Method

WcfDO.DocumentOnlineService.GetAttachDefintionMethod
You use this method to get the document attach definitions for the specified progID.

Parameters

records: (input) progID

returns: This returns an array of attachment definitions.

Data Contract

Name	Type	Description

AttachDefID	String	Gets or sets the ID of the attach def.
AttachObject	String	Gets or sets the attach object used for this attach definition.

Name	Type	Description
ProgID	String	Gets or sets the Prog. ID.
StorageObject	String	Gets or sets the storage object used for this attach definition.
Tables	Table	Gets or sets the fields used in the attach definition.

GetAttachedDocuments Method

WcfDO.DocumentOnlineService.GetAttachedDocumentMethod
You use this method to get the attached documents.

Parameters

records: (input) table - DocID

returns: DocumentInfoSummary array if documents are attached, otherwise null.

Data Contract - WcfDODocumentInfoSummary

Name	Type	Description
AttachDefID	String	Gets or sets the ID of the attach def.
AttachGuid	String	Gets or sets the attach object guide of the attachment.
AttachHistory Key	String	Gets or sets the attach history key of the attachment.
AttachID	String	Gets or sets the Attach ID of the document.
AttachInstance Key	String	Gets or sets the instance key of the attachment.
Create Date	DateTime	Gets or sets the create date of the attachment.
Description	String	Gets or sets the description.
DocID	Integer	Gets or set the Document ID.
Extension	String	Gets or sets the extension.
InstanceData	String	Gets or sets the instance data
StorageGuid	String	Gets or sets the storage object guid of the attachment.
URL	String	Gets or sets the url of the attachment.

GetDocumentURL Method

WcfDO.DocumentOnlineService.GetDocumentURLMethod
You use this method to get the document url.

Parameters

records: (input) DocID

returns: It returns Document url if successful, otherwise null.

GetThumbnails Method

WcfDO.DocumentOnlineService.GetThumbnailsMethod(WcfDO.DocumentInfoSummary[]) You use this method to get the thumbnails for the document specified.

Parameters

records:	(input) DocInfo - An array of document ID's to get thumbnails for.
returns:	An array of byte arrays, each byte array represents a thumbnail.

UploadandAttachMethod

WcfDO.DocumentOnlineService.UploadandAttachMethod
You use this method to upload a file and attach the file to a record.
.

Parameters

records:	(input) record, description, file ext, and data.
returns:	It returns document ID if successful, otherwise zero.

UploadFileMethod

WcfDO.DocumentOnlineService.UploadFileMethod
You use this method to upload a file to documents online.

Parameters

records:	(input) storage object, description, file ext, and data.
returns:	It returns document ID.

Encumbrances

InsertEnRecords

WcfEN.ENService.InsertEnRecords(WcfEN.Encumbrances[])
This method inserts, but does not validate an array of Encumbrance records. Removing validation from the process provides better performance when dealing with very large numbers of transactions.

Parameters

records:	(input) Array of EN records
returns:	True if no errors, otherwise false.

InsertAndValidateEnRecords

WcfEN.ENService.InsertAndValidateEnRecord(WcfEN.Encumbrances)
Identical to the previous service, this request processes a single record at a time, performing standard data validation prior to inserting the record.

Parameters

record:	(input) EN record
returns:	EN response for the inserted record

Data Contract - WcfEN.Encumbrances

Name	Type / Size	Description
BatchId	Char(16)	Set ID - Multiple records are grouped together in what is called a set or batch. If this field is blank, Finance Enterprise generates the batch ID. **Note**: We are currently transitioning from a maximum of 8 characters to 16 characters.
Ref	Char(16)	JE ID - Journal entry reference number. If this field is blank, Finance Enterprise generates the reference number.
PeId	Char(12)	PE ID - A Person/Entity ID relating to this entry.
RefDate	dateTime	Date - Transaction reference date. If this field is blank, Finance Enterprise adds the current date.
Desc	Char(30)	Description - Transaction description. The maximum length for this field is 30 characters.
GlGr	Char(2)	GL Ledger - General Ledger code for this entry. If this field is blank, Finance Enterprise uses the default. A finance code can also be used in this field.
GlKey	Char(10)	GL organization key - General ledger organization key. A finance code can be used in this field.

Name	Type / Size	Description
GlObj	Char(8)	GL object code - General Ledger object code for this entry. A finance code can also be used in this field.
JlGr	Char(2)	JL ledger - Job ledger code for this entry. If this field is blank, Finance Enterprise uses the user’s default code. A finance code can also be used in this field.
JlKey	Char(10)	JL organization key - Job ledger organization key for this entry. A finance code can also be used in this field.
JlObj	Char(8)	JL object code - Job ledger object code for this entry. A finance code can also be used in this field.
Wo	Char(12)	Work order - Work order number to be passed into the journal entry transaction if applicable.
Units	Num(20,5)	Units - Number of units with or without a debit or credit amount. This field is optional.
DistAmt	Num(20,2)	Debit - Positive debit or credit dollar amount. One debit or credit should be provided but never both.
Div	Char(4)	Division – Division code that allows for separation and grouping of transactions for reporting purposes.
FinCd	Char(8)	Finance code - Finance codes act as a link or shortcut to a financial account number, such as the org key or object code.

Fixed Asset

LookupAssetDetail

WcfFA.FAAssetCodesService. LookupAssetDetails (WcfFA.FADetailsLookupParameterFields)
This method returns all the asset details present in Finance for a particular asset ID and ledger.
Parameters
AssetLedger: Ledger (input)
AssetId: Asset ID (input)
returns: An array of fixed asset details which contain asset information, disposal information and appo information for a particular asset ID and ledger.

Data Contract - WcfFA.FADetailsLookupParameterFields

Name	Type / Size	Description
AssetId	string	Fixed asset ID
AssetDesc	string	Corresponding asset description
AssetStatus	string	Status of asset
AssetInServDt	DateTime	Asset in service date
AssetQty	Int	Number of quantities
AssetLedger	string	Ledger field
AssetPurchaseAmt	string	Purchase amount
DispMethod	string	Disposal method
DispDt	DateTime	Disposal date
DispSaleAmt	string	Disposal sale amount
FAAppoList[]	Array	This array contains list of fields mentioned below along with asset ID and ledger mentioned above.
AppoInvDt	DateTime	Appo related invoice date
AppoPo	string	Appo purchase order number
AppoContract	string	Appo contract number
AppoJlGr	string	JlGr account information
AppoJlObj	string	Object
AppoInvAmt	string	Appo invoice amount
AppoPeId	string	Person entity ID
AppoPeName	string	Person entity name

LookupAssetCode

WcfFA.FAAssetCodesService.LookupAssetCodes (Ledger, AssetType)
This method returns all the asset codes present in Finance for a particular asset type and ledger.
Parameters
Ledger: Ledger (input)
AssetType: Asset type – Primary Asset, Secondary Asset or Tertiary Asset (input)
returns: An array of fixed asset codes which contain ledger, code ID, code value and code description.

General Ledger – JE Service

InsertJeRecords

WcfGL.JEService.InsertJeRecords (WcfGL.JournalEntries[])
This method inserts, but does not validate an array of Accounts Receivable / Cash Receipt records. Removing validation from the process allows for better performance when dealing with very large numbers of transactions.
Parameters

records:	(input) Array of JE records
returns:	Array of JE responses for each of the record inserts

InsertAndValidateJeRecords

WcfGL.JEService.InsertAndValidateJeRecord (WcfGL.JournalEntries)
Identical to the previous service, this request processes a single record at a time, performing standard data validation prior to inserting the record.
Parameters
record: (input) An JE record
returns: An JE response for the inserted record

Data Contract - WcfGL.JournalEntries

Name	Type / Size	Description
BatchId	Char(16)	Set ID - Multiple records are grouped together in what is called a set or batch. If this field is blank, Finance Enterprise generates the set number. **Note**: We are currently transitioning from a maximum of 8 characters to 16 characters.
Ref	Char(16)	JE ID - Journal entry reference number. If this field is blank, Finance Enterprise automatically assigns the reference number.
Desc	Char(30)	Description - Transaction description. The maximum length for this field is 30 characters.
Date	dateTime	Date - Transaction reference date. If this field is blank, Finance Enterprise inserts the current date.
GlGr	Char(2)	GL ledger - General Ledger code for this entry. If this field is blank, the user's default code is used. A finance code can also be used in this field.
GlKey	Char(10)	GL organization key - General Ledger organization key for this entry. A finance code can also be used in this field.
GlObj	Char(8)	GL object code - General Ledger object code for this entry. A finance code can also be used in this field.
JlGr	Char(2)	JL ledger - Job ledger code for this entry. If this field is blank, the user's default code is used. A finance code can also be used as the default for this field.
JlKey	Char(10)	JL organization key - Job ledger organization key for this entry. A finance code can also be used for this field.
JlObj	Char(8)	JL object code - Job ledger object code for this entry. A finance code can be used in this field. When the Finance Code is used that these fields can be left blank.
Wo	Char(12)	Work order - Work order number that is passed into the journal entry transaction if needed.
Dr	Num(20,2)	Debit - Positive debit or credit dollar amount. Provide one debit or credit, but not both.
Cr	Num(20,2)	Credit - Positive debit or credit dollar amount. One debit or credit should be provided but never both.
Units	Num(20,5)	Units - Number of units. This field is optional and can be provided with or without a debit or credit amount.
OffsetIntra	Char(1)	Intrafund offset - Set to **Y** to apply intrafund offsets automatically when distributing this transaction.
OffsetInter	Char(1)	Interfund offset - Set to **Y** to apply interfund offsets automatically when distributing this transaction.
Ref2	Char(16)	Secondary reference - Reference number related to the origin or subsystem of the source transaction
PeId	Char(12)	PE ID - Person/Entity ID relating to this entry.
ContractNo	Char(16)	Contract no - Contract number that relates to this entry.

Name	Type / Size	Description
PrepId	Char(8)	Prepared by - User ID of the person who prepared this entry.
Misc	Char(4)	Misc - User-defined code that identifies the transaction for reporting purposes.
HitEn	Char(2)	Hit other - Subsystem to which this transaction should be entered: Value Purpose N General Ledger only (default) Y Encumbrances and General Ledger E Encumbrances only B Budgets only K Bank Reconciliation and General Ledger L Bank Reconciliation only O Accounts Payable open-hold only I Accounts Payable immediate pay only

Name	Type / Size	Description
EnType	Char(2)	Other type - If the subsystem field **HitEn** is not equal to **N**, a transaction type must be in this field. The following transaction types are allowed: Accounts Payable or Encumbrances EN - Encumbrance DE - Disencumbered PP - Partial Payment FP - Full Payment Bank Reconciliation AJ - Adjustment BF - Fee BI - Interest CK - Check DP - Deposit EF - Electronic Funds Transfer RV - Reversal Budgets A - Budget Adjustment N - New Budget
Date2	dateTime	Secondary date - Subsystem reference date.
CkId	Char(2)	Check ID - Check stock ID or bank ID used with this entry. This code must be defined in Nucleus Common Codes file under the code category **CKID**. This field is required if **HitEn** is equal to **K** or **L**.
CkNo	Char(8)	Check number - Check number that relates to this entry. This field is required if **HitEn** is **K** or **L**.
FinCd	Char(8)	Finance code - Finance codes act as a link or shortcut to a financial account number, such as the org key or object code. Including a pre-defined finance code allows the following six properties to be left blank – G1Gr, G1Key, G1Obj, J1Gr, J1Key, J1Obj.

General Ledger – Account Lookup Service

AccountLookupMethod

It does a lookup the account that matches the criteria specified.
WcfGL.AccountLookupservice.AccountLookup

Parameters

Records: Max Rows, ledger, year, and parameters
Returns: An array of records containing the account and additional information.

Data Contract – Account Lookup Record

Name	Type	Description
Account	String	The account.
Ledger	String	The Ledger
ObjBalType	String	The balance type of the object code, CR or DR.
Object	String	The Object code.
ObjGroup1 to ObjGroup8	String	Group1 – 8 of object code.
ObjLongDescription	String	The long description of the object code.
ObjShortDescription	String	The short description of the object code.
ObjStatus	String	The status of the object code.
ObjType	String	The account type of the object code.
OrgKey	String	The organization key
OrgKeyLongDesc	String	The long description of the org key.
OrgKeyPart1 to OrgKeyPart8	String	The background key 1 to 8 of the organization key.
OrgKeyShortDesc	String	The short description of the org key.
OrgKeyStatus	String	The status of the org key.
Year	String	The fiscal year.

GetAccountsMethod

WcfGL.AccountLookupservice.GetAccountsMethod (WcfGl.AccountLookupRequest) It does a lookup for the accounts that matches the criteria specified.

Parameters

Record: Request (input)
Returns: An array of records containing the account and additional information.

Data Contract – Account Lookup Request

Name	Type	Description
Fiscal Year	Integer	The account.
IncludeNon Transactional	Boolean	Include the NonTransactional or Budget type accounts.
Ledger	String	The Ledger
MaxRows	Integer	Maximum number of rows to return. There is no limit on rows when null or undefined.
Parameters	Array	An array of Lookup Parameters.
SortParameters	Array	An array of Sorting Parameters.

GetKeysMethod

WcfGL.AccountLookupservice.GetKeysMethod (WcfGl.KeyLookupRequest)
It does a lookup for the object codes that match the criteria specified.
Parameters
Record: Request (input)
Returns: An array of records containing the account and additional information.

Data Contract – Keylookup Request

Name	Type	Description
Ledger	String	The ledger associated with the org key.
OrgKey	String	The Org Key.
OrgKeyEndDate	String	The end date of org key.
OrgKey Part1 to Part 8	String	Background 1 – 8 of org key.
OrgKey Long Description	String	The long description of the org key.
OrgKeyShortDescription	String	The short description of the org key.
OrgKeyStartDate	String	The start date of organization key.
OrgKeyStatus	String	The status of the organization key.

GetObjectsMethod

WcfGL.AccountsLookupSevice.GetObjectMethod (WcfGL.ObjectLookupRequest)
You use this method to lookup for the object codes that match the criteria specified.
Parameters
records: (input) Request parameters

returns: An array of records containing the account and additional information.

Data Contract – ObjectLookup Record

Name	Type	Description
Ledger	String	The ledger.
ObjBalType		The balance type of the object code, CR or DR.
Object		The object code.
ObjEndDt	String	The date which is the last date that the object code is valid.
ObjGroup1- ObjGroup8	String	Group Code 1 - 8 of object key.
ObjLong Description	String	The long description of the object.
ObjShortDescription	String	The short description of the object.
ObjStartDt	String	The date on which the object code becomes valid.
ObjType	String	The account type of the object code.
ObjStatus	String	The status of the object.

KeyLookupMethod

WcfGL.AccountsLookupSevice.KeyLookupMethod(WcfGL.KeyLookupRequest)
You use this method to lookup the org keys that match the criteria specified.
.

Parameters

records: (input) max rows, ledger, Parameters
returns: An array of records containing the account and additional information.

ObjectLookupMethod

WcfGL.AccountsLookupSevice.ObjectLookupMethod (WcfGL.ObjectLookupRequest)
You use this method to lookup the object codes that match the criteria specified.

.

Parameters

records: (input) max rows, ledger, Parameters
returns: An array of records containing the account and additional information.

General Ledger – Account Specific Service

GetActiveGLJLAccounts Method

Use this to get active GLJL Accounts.

WcfGL.AccountSpecificservice.GetActiceGLJLAccounts method (WcfGL.GLJLAccountsRequest)
Records: Request (input)
Returns: GLJL Accounts Response.

Data Contract – GLJLAccounts Response

Name	Type / Size	Description
Key	String	The key associated with the account.
Ledger	String	The ledger associated with the account.
ObjDl	String	The object Dl.
Object	String	This describes the object.
Title DL	String	This describes the title DL.

GetGLFiscalInformationMethod

Use this to get GL account fiscal information.
WcfGL.GLAccountSpecificservice.GeticeGLFiscalInformationMethod (WcfGL.GetGLFiscalInformation)
Records: Request (input)
Returns: Fiscal Period Year response.

Data Contract – GLJLFiscalInformationResponse

Name	Type / Size	Description
Fiscal Year	String	This describes the fiscal year.
Fiscal Period	String	This describes the fiscal period.

GetGLJLAccountsMethod

Use this to get GLJL account information.
WcfGL.GLAccountSpecificServices.GetGLJLAccounts Method (WcfGL. GLJLAccountsRequest)
Records: Request (input)
Returns: GLJL Accounts Response.

Data Contract – GLJLAccountResponse

Name	Type / Size	Description
Key	String	The key associated with the account.
Ledger	String	The ledger associated with the account.
ObjDl	String	The object Dl.
Object	String	This describes the object.
Title DL	String	This describes the title DL.

Data Contract – GLJLAccountRequest

Name	Type / Size	Description
Effective Date	DateTime	This describes the effective Date when used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
Fiscal Year	String	This describes the fiscal year.
Ledger	String	The ledger associated with the account.
MaxRows	Integer	This describes the max rows.
Type	String	This describes the type.

GetGLJLCodesMethod

Use this to get GLJL codes.
WcfGL.GLAccountSpecificServices.GetGLJLCodes Method (WcfGL. GLJLCodesRequest)
Records: Request (input)
Returns: GLJL Codes Response.

Data Contract – GLJLCodesResponse

Name	Type / Size	Description
Desc	String	This describes the description.
FiscMo		
JlList	String	
Ledger	String	
Title	String	
Type	String	

Data Contract – GLJLCodesRequest

Name	Type / Size	Description
Ledger	String	This describes the ledger.
Type	String	This describes the type.

GetGLJLKeysMethod

Use this to get GLJL keys.
WcfGL.GLAccountSpecificServices.GetGLJLKeys Method (WcfGL. GLJLKeysRequest)
Records: Request (input)
Returns: GLJL Keys Response.

Data Contract – GLJLKeysResponse

Name	Type / Size	Description
Key	String	This describes the keys.
Ledger	String	This describes the ledger.

Name	Type / Size	Description
TitleDL	String	This describes the title DL.

Data Contract – GLJLKeysRequest

Name	Type / Size	Description
AcceptTr	String	This describes the accept transactions.
Effective Date	DateTime	This describes the effective date when used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
Key	String	This describes the key.
Ledger	String	This describes the ledger.
Status	String	This describes the status.

GetGLJLObjectsMethod

Use this to get GLJL objects.
WcfGL.GLAccountSpecificServices.GetGLJLObjects Method (WcfGL. GLJLObjectsRequest)
Records: Request (input)
Returns: GLJL Objects Response.

Data Contract – GLJLObjectResponse

Name	Type / Size	Description
Ledger	String	This describes the ledger.
ObjDL	String	This describes the ObjDL.
Object	String	This describes the object.

Data Contract – GLJLObjectRequest

Name	Type / Size	Description
AllowSubs08	String	AllowSubs08
AllowSubs09	String	AllowSubs09
Effective Date	Date Time	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
Ledger	String	This describes the ledger.
Object	String	This describes the object.
Status	String	This describes the status.

General Ledger – Budgeting Service

GetBudgetBalanceMethod

Use this method to look up the budget available for a fiscalyear/ledger/key/object/level combination out of GL Level is always "OB."
WcfGL.GLBudgetingServices.GetBudgetBalance Method (WcfGL.GetBudgetRequest)
Records: Budget Request (input) - The search criteria to find budget amount. Returns: The outstanding balance for the budget search specified.

Data Contract – GetBudgetResponse

Name	Type / Size	Description
Actual Total	Decimal	This describes the total amount of budget actualized (used) in fiscal year.
Budget Amount	Decimal	This describes the budget amount for the fiscal year.
Budget Balance	Decimal	This describes the budget balance calculated with formula below BudgetAmount - (EncumberedTotal + ActualTotal.)
Encumbered Total	Decimal	This describes the total amount encumbered against the budget in fiscal year.

General Ledger – Validation Service

ValidateGLAccountMethod

Use this method to perform the General Ledger / Job Ledger account number validation when passing individual account number components.
WcfGL.GLValidationService.ValidateGLAccount Method (WcfGL.GLValidationRequest) Records: GL Account Request (input) - The GL account validation request information. Returns: The results of the GL validation.

Data Contract – GLValidationResponse

Name	Type / Size	Description
GL Key Description	String	The description of the GL key.
GL Object Description	String	The description of the GL object.
JL Key Description	String	The description of the JL key.
JL Object Description	String	The description of the JL object.
Object Type	String	This describes the object type.
Work Order Description	String	The description of the work order.

Data Contract – GLValidationRequest

Name	Type / Size	Description
GL Key	String	The GL key to validate.

Name	Type / Size	Description
GL Ledger Code	String	The GL ledger to validate against.
GL Object Code	String	The GL object code to validate.
JL Key	String	The JL key to validate.
JL Ledger Code	String	The JL ledger code to validate.
JL Object Code		The JL object code to validate against.
Work Order	String	The work order to validate.

ValidateGLAccountStringMethod

Use this method to perform the General Ledger / Job Ledger account number validation when passing in an account string, which will first be parsed for individual components.
WcfGL. GLValidationService.ValidateGLAccountString Method (WcfGL.AccountType)
Records: (input)

- Account String - The account string describing the account number
- GL Ledger Code – The GL Ledger
- JL Ledger Code – The JL Ledger
- Account Type – The account type

Returns: The results of the GL validation.

Data Contract – GLValidationResponse

Name	Type / Size	Description
GL Key Description	String	The description of the GL key.
GL Object Description	String	The description of the GL object.
JL Key Description	String	The description of the JL key.
JL Object Description	String	The description of the JL object.
Object Type	String	This describes the object type.
Work Order Description	String	The description of the work order.

ValidateNLProjectMethod

Use this method to perform a special check for a NaviLine project. For use with interfaces developed specifically between NaviLine and OS Finance products.
WcfGL. GLValidationService.ValidateNLProject Method (WcfGL.GLValidationRequest) Records:
GL Account Request (input) – The GL account validation request information. Returns: The results of the GL validation.

Data Contract – GLValidationResponse

Name	Type / Size	Description
GL Key Description	String	The description of the GL key.
GL Object Description	String	The description of the GL object.
JL Key Description	String	The description of the JL key.
JL Object Description	String	The description of the JL object.
Object Type	String	This describes the object type.
Work Order Description	String	The description of the work order.

Data Contract – GLValidationRequest

Name	Type / Size	Description
GL Key	String	The GL key to validate.
GL Ledger Code	String	The GL ledger to validate against.
GL Object Code	String	The GL object code to validate.
JL Key	String	The JL key to validate.
JL Ledger Code	String	The JL ledger code to validate.
JL Object Code		The JL object code to validate against.
Work Order	String	The work order to validate.

Human Resources – Adhoc Request Services

DeleteAdhocRequestMethod

Use this method to delete AdhocRequest record given a RequestID.
WcfHR.AdhocRequestServices.DeleteAdhocRequest Method
Records: Request ID (input)
Returns: True if delete is successful, otherwise false.

GetOpenPafRequestsMethod

Use this method to GetOpenPafRequests.
WcfHR.AdhocRequestServices.GetOpenPafRequests Method (WcfHR.OpenPafRequest)
Records: Request (input)
Returns: Row Count
Data Contract – OpenPafRequest

Name	Type / Size	Description
Adhoc Action Type	String	This describes the adhoc action type.
Adhoc Employee ID	String	This describes the adhoc employee ID.

GetSeedFromCodeValueMethod

Use this method to get seed from code value.
WcfHR.AdhocRequestServices.GetSeedFromCodeValue Method
Records: Code Value (input) - Code Value used to generate seed.
Returns: Next seed if no errors, otherwise blank.

InsertAndValidateAdhocRequestMethod

Use this method to inserts a single Adhoc Request record with rule validation.
WcfHR.AdhocRequestServices.InsertAndValidateAdhocRequest Method
(WcfHR.AdhocRequestSubset)
Records: Request (input) - LookupAdhocRequestRequest object.
Returns: Valid Insert Order Response if no errors, otherwise null.
Data Contract – InsertAdhocRequestResponse

Name	Type / Size	Description
RequestId	String	This describes the RequestId Property.

Data Contract – AdhocRequestSubset Properties

Name	Type / Size	Description
Action Type	String	This describes ActionType Property.
Comments	String	This describes Comments Property.
Create When	DateTime	This describes CreateWhen Property.
Create Who	String	This describes Create Who Property.
DeptLongDesc	String	This describes DeptLongDesc Property (HRHrencode table.)
DivLongDesc	String	This describes DivLongDesc Property (HRHrencode table.)
Effective Date	DateTime	This describes EffectiveDate Property.
EmpDepartment	String	This describes EmpDepartment Property (HREmpmstr table.)
EmpDivision	String	This describes EmpDivision Property (HREmpmstr table.)

Name	Type / Size	Description
EmployeeId	String	This describes EmployeeId Property.
EmpName	String	This describes EmpName Property (HREmpmstr table.)
ExpireDt	DateTime	This describes ExpireDt Property.
Reason	String	This describes Reason Property.
RequestId	String	This describes RequestId Property.
Requestor	String	This describes RequestorId Property.
SeedCode	String	This describes Seed Code When RequestId is blank, a seed value is generated using a common code with code category = SYNO and code value = SeedCode.
Status Code	String	This describes StatusCode Property.
Unique ID	String	This describes UniqueId Property.
Unique Key	String	This describes UniqueKey Property.
Update When	DateTime	This describes UpdateWhen Property.
Update Who	String	This describes UpdateWho Property.

LookupAdhocRequestRecords Method

Use this method to return an adhoc request record or records based on the input parameters.
WcfHR.AdhocRequestServices. LookupAdhocRequestRecords Method (WcfHR. LookupAdhocRequestRequest)
Records: Request (input) - A LookupAdhocRequestRequest object.
Returns: A LookupAdhocRequestResponse.

Data Contract – LookupAdhocRequestResponse

Name	Type / Size	Description
Adhoc Request Array		Array of AdhocRequest records

Data Contract – LookupAdhocRequestRequest

Name	Type / Size	Description
MaxRows		Maximum number of rows to return No limit on rows when null or undefined.
OrderByFields		Order By Parameter Field
Parameters		Array of AdhocRequest Lookup Parameters

Name	Type / Size	Description
SelectDistrict		Select Distinct
SelectFields		Array of Select Fields

LookupAdhocRequestWithADHOCTrns Method

Lookup AdhocRequest with ADHOCTrns table.
WcfHR.AdhocRequestServices.LookupAdhocRequestWithADHOCTrns Method (WcfHR. LookupAdhocRequestWithJoinRequest)
Records: Request (input) - LookupAdhocRequestWithJoinRequest.
Returns: LookupAdhocRequestResponse.

Data Contract – LookupAdhocRequestResponse

Name	Type / Size	Description
Adhoc Request Array		Array of AdhocRequest records

Data Contract – LookupAdhocRequestWithJoinRequest

Name	Type / Size	Description
Entity ID	String	This describes the entity ID.
Max Rows		This describes the maximum number of rows to return No limit on rows when null or undefined.
Parameters		This describes an array of AdhocRequest Lookup parameters.

LookupAdhocRequestWithHREmpmstr Method

Lookup AdhocRequest with HREmpmstr table.
WcfHR.AdhocRequestServices.LookupAdhocRequestWithHREmpmstrMethod (WcfHR.LookupAdhocRequestWithJoinRequest) Records: Request (input) - LookupAdhocRequestRequest.
Returns: LookupAdhocRequestResponse.

Data Contract – LookupAdhocRequestResponse

Name	Type / Size	Description
Adhoc Request Array		Array of AdhocRequest records

Data Contract – LookupAdhocRequestWithJoinRequest

Name	Type / Size	Description
Entity ID	String	This describes the entity ID.
Max Rows		This describes the maximum number of rows to return No limit on rows when null or undefined.
Parameters		This describes an array of AdhocRequest Lookup parameters.

SearchAdhocRequests Method

Use this method to Search Adhoc requests.
WcfHR.AdhocRequestServices.LookupAdhocRequestWithHREmpmstrMethod (WcfHR.SearchAdhocRequestsMethod)
Records: Request (input) - Search Adhoc Request.
Returns: Search Adhoc Response.

Data Contract – SearchAdhocResponse

Name	Type / Size	Description
AdhocActionType	String	AdhocActionType
AdhocComments	String	AdhocComments
AdhocEffectiveDt	DateTime	AdhocEffectiveDt
AdhocEmployeeId	String	AdhocEmployeeId
AdhocExpireDt	DateTime	AdhocExpireDt
AdhocReason	String	AdhocReason
AdhocRequestId	String	AdhocRequestId
AdhocRequestor	String	AdhocRequestor
AdhocStatusCd	String	AdhocStatusCd
CreateWhen	DateTime	CreateWhen
CreateWho	String	CreateWho
DeptCodeval	String	DeptCodeval
DeptLongDesc	String	DeptLongDesc
DivCodeval	String	DivCodeval
DivLongDesc	String	DivLongDesc
UniqueKey	String	UniqueKey

Name	Type / Size	Description
UpdateWhen	DateTime	UpdateWhen
UpdateWho	String	UpdateWho

Data Contract – SearchAdhocRequest

Name	Type / Size	Description
Entity ID	String	This describes the entity ID.
RequestId	String	This describes the request Id.
SkipTableJoin		Skip table join with with HREmpmstr and Hrencode tables.
UniqueKey		This describes the unique key.

UpdateAndValidateAdhocRequest Method

Use this method to update a single Adhoc Request record with rule validation.
WcfHR.AdhocRequestServices. UpdateAndValidateAdhocRequest Method (WcfHR. AdhocRequestSubset)
Records: Request (input) - Search Adhoc Request.
Returns: UpdateAdhocRequestResponse object.

Data Contract – UpdateAdhocRequestResponse

Name	Type / Size	Description
RequestId	String	This describes the request Id property.

Data Contract – AdhocRequestSubset

Name	Type / Size	Description
ActionType	String	ActionType Property
Comments	String	Comments Property
CreateWhen	DateTime	CreateWhen Property
CreateWho	String	CreateWho Property
DeptLongDesc	String	DeptLongDesc Property (HRHrencode table)
DivLongDesc	String	DivLongDesc Property (HRHrencode table)
EffectiveDate	DateTime	EffectiveDate Property
EmpDepartment	String	EmpDepartment Property (HREmpmstr table)
EmpDivision	String	EmpDivision Property (HREmpmstr table)

Name	Type / Size	Description
EmployeeId	String	EmployeeId Property
EmpName	String	EmpName Property (HREmpmstr table)
ExpireDt	DateTime	ExpireDt Property
Reason	String	Reason Property
RequestId	String	RequestId Property
Requestor	String	RequestorId Property
SeedCode	String	Seed Code When RequestId is blank, a seed value is generated using a common code with code category = SYNO and code value = SeedCode.
StatusCode	String	StatusCode Property
UniqueId	String	UniqueId Property
UniqueKey	String	UniqueKey Property
UpdateWhen	DateTime	UpdateWhen Property
UpdateWho	String	UpdateWho Property

Human Resources – Codes Services

LookupBargainingUnitCodes Method

Use this service method to lookup Bargaining Unit Codes records based upon requested search criteria.
WcfHR.HRCodesServices.LookupBargainingUnitCodes Method (WcfHR.LookupBargainingUnitCodesRequest)
Records: Request (input) - The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.
Returns: Returns an array of HrHrencodeSubset records based upon the search criteria, otherwise returns null.
Data Contract – LookupBargainingUnitCodesResponse

Name	Type / Size	Description
HrBargtble Array		Array of HrBargtbleSubset records

Data Contract – LookupBargainingUnitCodesRequest

Name	Type / Size	Description
MaxRows		Maximum number of rows to return No limit on rows when null or undefined
OrderByFields		Order By Parameter Field
Parameters		Array of Codes Lookup Parameters
SelectDistinct		Select Distinct
SelectFields		Select Fields

LookupEntitySpecificCodes Method

Use this service method to lookup Entity Specific Codes records based upon requested search criteria.
WcfHR.HRCodesServices.LookupEntitySpecificCodesRequest (WcfHR. LookupEntitySpecificCodesRequest)
Records: Request (input) - The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.
Returns: Returns an array of HrHrencodeSubset records based upon the search criteria, otherwise returns null.

Data Contract – LookupEntitySpecificCodesResponse

Name	Type / Size	Description
HrHrencode Array		Array of HrHrencodeSubset records

Data Contract – LookupEntitySpecificCodesRequest

Name	Type / Size	Description
MaxRows		Maximum number of rows to return No limit on rows when null or undefined
OrderByFields		Order By Parameter Field
Parameters		Array of Codes Lookup Parameters
SelectDistinct		Select Distinct
SelectFields		Select Fields

LookupNonEntitySpecificCodes Method

Use this service method to lookup Non Entity Specific Codes records based upon requested search criteria.
WcfHR.HRCodesServices.LookupNonEntitySpecificCodesMethod (WcfHR. LookupNonEntitySpecificCodesRequest)

Records: Request (input) - The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

Returns: Returns an array of HrHrcodeSubset records based upon the search criteria, otherwise returns null.

Data Contract – LookupNonEntitySpecificCodesResponse

Name	Type / Size	Description
HrHrcodeArray		Array of HrHrcodeSubset records

Data Contract – LookupNonEntitySpecificCodesRequest

Name	Type / Size	Description
MaxRows		Maximum number of rows to return No limit on rows when null or undefined
OrderByFields		Order By Parameter Field
Parameters		Array of Codes Lookup Parameters
SelectDistinct		Select Distinct
SelectFields		Select Fields

Human Resources – Employee Master Services

GetEmpData Method

Use this REST service to get information about Employees.

Name	Type / Size	Description
	String	[Optional] The Employee ID if running for a single employee.

Query string options:

SkipRows: The number of rows at the beginning of the query to not report.

TakeRows: The number of rows from the query to report. If left blank the default is 1000.

?TakeRows=10 results in only 10 rows being returned.

?SkipRows=10 results in all rows after row 10 being returned.

?SkipRows=10&TakeRows=10 results in rows 11 through 20 being returned.

Fields: A list of table columns to be included in the query. If this is included in the query string, then only the listed columns will be reported.

?fields=fname,lname results in only the first and last name being reported.

Columns can be renamed by appending the new name after the default name.

?fields=fname+FirstName,lname+LastName

Filters:

Any column name can be added to the query string as a filter. For example

?fname=Dan

?hdt=1/1/2019

By default the filter comparison is equality. Other comparisons are available and can be appended to the filter

"?fname=Dan+like " will select Dan, Daniel and Dannie

"?hdt=1/1/2019+gt" will include all dates after 1/1/2019

Available comparisons are:

"ge", greater than or equal to (>=)

"gt", greater than (>)

"isnotnull", is not null

"isnull", is null

"le", less than or equal to (<=)

"like", like

"lt", less than (<)

"ne", not equal (!= , <>)

"notlike", not like

GetEmpPCNData

Use this REST service to get information about employees (including position information).

Query string options:

SkipRows: The number of rows at the beginning of the query to not report.

TakeRows: The number of rows from the query to report. If left blank the default is 1000.

?TakeRows=10 results in only 10 rows being returned.

?SkipRows=10 results in all rows after row 10 being returned.

?SkipRows=10&TakeRows=10 results in rows 11 through 20 being returned.

Fields: A list of table columns to be included in the query. If this is included in the query string, then only the listed columns will be reported.

?fields=fname,lname results in only the first and last name being reported.

Columns can be renamed by appending the new name after the default name.

?fields=fname+FirstName,lname+LastName

Filters:

Any column name can be added to the query string as a filter. For example:

?fname=Dan

?hdt=1/1/2019

By default, the filter comparison is equality. Other comparisons are available and can be appended to the filter.

"?fname=Dan+like " will select Dan, Daniel and Dannie

"?hdt=1/1/2019+gt" will include all dates after 1/1/2019

Available comparisons are:

"ge", greater than or equal to (>=)

"gt", greater than (>)

"isnotnull", is not null

"isnull", is null

"le", less than or equal to (<=)

"like", like

"lt", less than (<)

"ne", not equal (!= , <>)

"notlike", not like

LookupEmployeeMaster Method

Use this service method to lookup Employee Master records based upon requested search criteria.
WcfHR.HREmployeeMasterServices.LookupEmployeeMaster Method
(WcfHR.LookupEmployeeMasterRequest)
Records: Request (input) - The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.
Returns: Returns an array of HrEmpmstrSubset records based upon the search criteria, otherwise returns null.

Data Contract – LookupEmployeeMasterResponse

Name	Type / Size	Description
HrEmpmstr Arrayy		Array of HrEmpmstrSubset records

Data Contract – LookupEmployeeMasterRequest

Name	Type / Size	Description
EffectiveDate	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
MaxRows		Maximum number of rows to return No limit on rows when null or undefined

Name	Type / Size	Description
OrderByFields		Order By Parameter Field
Parameters		Array of Codes Lookup Parameters
SelectDistinct		Select Distinct
SelectFields		Select Fields

Human Resources – Employee Salary Services

GetEmployeeSalariesFromScheduleAndRange Method

Use this service method to get Employee Salaries From Schedule And Range.
WcfHR. HREmployeeSalaryServices.GetEmployeeSalariesFromScheduleAndRange Method (WcfHR. EmployeeSalariesFromScheduleAndRangeRequest) Records: Request (input) - Salary Information Request. Returns: Salary Information Response.

Data Contract – SalaryInformationResponse

Name	Type / Size	Description
Amount	String	Amount
Axp	String	Axp
IndxKey	String	Index Key (Schedule/Range/Step)
Payeffbeg	DateTime	Pay effective beginning
Payeffend	DateTime	Pay effective end
Range	String	Range
SchedType	String	Schedule Type
Schedule	String	Schedule
Step	String	Step
StepDesc	String	StepDesc

Data Contract – EmployeeSalariesFromScheduleAndRangeRequest

Name	Type / Size	Description
EffectiveDate	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
EntityId	String	Entity Id

Name	Type / Size	Description
Range	String	Range
Schedule	String	Schedule

GetSalaryInformation Method

Use this service method to get Get Salary Information.
WcfHR. HREmployeeSalaryServices.GetSalaryInformation Method (WcfHR. SalaryInformationRequest)
Records: Request (input) - Salary Information Request.
Returns: Salary Information Response.

Data Contract – SalaryInformationResponse

Name	Type / Size	Description
Amount	String	Amount
Axp	String	Axp
IndxKey	String	Index Key (Schedule/Range/Step)
Payeffbeg	DateTime	Pay effective beginning
Payeffend	DateTime	Pay effective end
Range	String	Range
SchedType	String	Schedule Type
Schedule	String	Schedule
Step	String	Step
StepDesc	String	StepDesc

Data Contract – SalaryInformationRequest

Name	Type / Size	Description
EffectiveDate	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
EntityId	String	Entity Id
IndxKey	String	Index Key (Schedule/Range/Step)

LookupSalaryDetail Method

Use this service method to lookup salary table records based upon requested search criteria.
WcfHR. HREmployeeSalaryServices.LookupSalaryDetail Method (WcfHR. LookupSalaryDetailRequest)

Records: Request (input) - The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.
Returns: Returns an array of HrSlrytbleSubset records based upon the search criteria, otherwise returns null.

Data Contract – LookupSalaryDetailResponse

Name	Type / Size	Description
HrSlrytbleArray		Array of HrSlrytbleSubset records

Data Contract – LookupSalaryDetailRequest

Name	Type / Size	Description
MaxRows		Maximum number of rows to return No limit on rows when null or undefined
OrderByFields		Order By Parameter Field
Parameters		Array of Codes Lookup Parameters
SelectDistinct		Select Distinct
SelectFields		Select Fields

Human Resources – Pay Assignments Services

GetAllStatusAdjustmentRecType Method

Use this service method to get all Status Adjustment RecType.
WcfHR.HRPayAssignmentsServices.GetAllStatusAdjustmentRecType Method (WcfHR.StatusAdjustmentRecTypeRequest)
Records: Request (input) - Status Adjustment RecType Request.
Returns: Status Adjustment RecType Response.

Data Contract – StatusAdjustmentRecTypeResponse

Name	Type / Size	Description
EmpUnique Key	String	Emp Unique Key (HREmpmstr table)
PayUnique Key	String	Pay Unique Key (HREmppay table)
PcnFte	String	PcnFte
Pcn Long Description	String	Pcn Long Description
PcnPosLong	String	Pcn Position Long
PcnShortDesc	String	Pcn Short Description

Name	Type / Size	Description
PcnUnique Key	String	Pcn Unique Key (HRPcntble table)
Position	String	Position
RecType	String	Record Type

Data Contract – StatusAdjustmentRecTypeRequest

Name	Type / Size	Description
EffectiveDa te	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
EmployeeId	String	This describes the employee ID.
EntityId	String	This describes the entity ID.
RecordType	String	Record Type

GetCurrentlyAssignedAmtPerPcn Method

Use this service method to get currently assigned AmtPerPcn.
WcfHR.HRPayAssignmentsServices.GetCurrentlyAssignedAmtPerPcn Method (WcfHR. CurrentlyAssignedSumPerPcnRequest)
Records: Request (input) - CurrentlyAssignedSumPerPcnRequest.
Returns:

Data Contract – CurrentlyAssignedSumPerPcnRequest

Name	Type / Size	Description
EffectiveDa te	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
Pcn	String	Pcn

GetCurrentlyAssignedFtePerPcn Method

Use this service method to get currently assigned FtePerPcn.
WcfHR.HRPayAssignmentsServices.GetCurrentlyAssignedFtePerPcn Method (WcfHR. CurrentlyAssignedSumPerPcnRequest)
Records: Request (input) - CurrentlyAssignedSumPerPcnRequest.
Returns:

Data Contract – CurrentlyAssignedSumPerPcnRequest

Name	Type / Size	Description
EffectiveDa te	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
Pcn	String	Pcn

GetStatusAdjustmentRecType Method

Use this service method to get status adjustment RecType.
WcfHR.HRPayAssignmentsServices.GetStatusAdjustmentRecType Method (WcfHR.
StatusAdjustmentRecTypeRequest)
Records: Request (input) - Status Adjustment RecType Request.
Returns: Status Adjustment RecType Response

Data Contract – StatusAdjustmentRecTypeResponse

Name	Type / Size	Description
EmpUnique Key	String	Emp Unique Key (HREmpmstr table)
PayUnique Key	String	Pay Unique Key (HREmppay table)
PcnFte	String	PcnFte
Pcn Long Description	String	Pcn Long Description
PcnPosLong	String	Pcn Position Long
PcnShortDesc	String	Pcn Short Description
PcnUnique Key	String	Pcn Unique Key (HRPcntble table)
Position	String	Position
RecType	String	Record Type

Data Contract – StatusAdjustmentRecTypeRequest

Name	Type / Size	Description
EffectiveDate	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
EmployeeId	String	This describes the employee ID.
EntityId	String	This describes the entity ID.
RecordType	String	Record Type

LookupPayAssignments Method

Use this service method to lookup Pay Assignments records based upon requested search criteria.
WcfHR.HRPayAssignmentsServices.LookupPayAssignments Method (WcfHR.
LookupPayAssignmentsRequest)
Records: Request (input) - The request includes the maximum number of rows to return and an array of parameteters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

Returns: Returns an array of HrEmppaySubset records based upon the search criteria, otherwise returns null.

Data Contract – LookupPayAssignmentsResponse

Name	Type / Size	Description
HrEmppay Array		Array of HrEmppaySubset records

Data Contract – LookupPayAssignmentsRequest

Name	Type / Size	Description
EffectiveDate	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
MaxRows		Maximum number of rows to return No limit on rows when null or undefined
OrderByFields		Order By Parameter Field
Parameters		Array of Codes Lookup Parameters
SelectDistinct		Select Distinct
SelectFields		Select Fields

Human Resources – Position Definition Services

LookupPositionDefinitions Method

Use this service method to lookup position definitions records based upon requested search criteria.
WcfHR.HRPositionDefinitionsServices.LookupPositionDefinitions Method (WcfHR.
LookupPositionDefinitionsRequest)
Records: Request (input) - The request includes the maximum number of rows to return and an array of parameteters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.
Returns: Returns an array of HrPcntbleSubset records based upon the search criteria, otherwise returns null.

Data Contract – LookupPositionDefinitionsResponse

Name	Type / Size	Description
HrPcntbleArray		Array of HrPcntbleSubset records

Data Contract – LookupPositionDefinitionsRequest

Name	Type / Size	Description
EffectiveDate	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
MaxRows		Maximum number of rows to return No limit on rows when null or undefined
OrderByFields		Order By Parameter Field

Name	Type / Size	Description
Parameters		Array of Codes Lookup Parameters
SelectDistinct		Select Distinct
SelectFields		Select Fields

Human Resources – Position Specific Services

GetAllPositionSearch Method

Use this service method to search for position definitions records based upon requested search criteria.
WcfHR.HRPositionSpecificServices.GetAllPositionSearchMethod (WcfHR. PositionSearchRequest)
Records: Request (input) – Postion search request
Returns: Returns position search resposne.

Data Contract – PositionSearchResponse

Name	Type / Size	Description
AssgNo.	String	This displays the assignement number.
Job Code	String	This dispalsy the job code.
JobLong Description	String	This dispalsy the job long description.
JobShortDescription	String	This dispalsy the job short description.
Location	String	This dispalsy the location.
LocLongDesc	String	This displays the location long description.
LocShortDesc	String	This displays the location short description.
PcnLongDesc	String	This displays the position long description.
PcnShortDesc	String	This displays the position short description.
PcnPosLong	String	This displays the pcn position long.
Position	String	This displays the position.

Data Contract – PositionSearchRequest

Name	Type / Size	Description
EntityID	String	This displays the Entity ID.
PositionID	String	This displays the position ID.

GetAllPositionSearch Method

Use this service method to search for position definitions records based upon requested search criteria.
WcfHR.HRPositionSpecificServices.GetAllPositionSearchMethod (WcfHR. PositionSearchRequest)
Records: Request (input) – Position search request
Returns: Returns position search response.
Data Contract – PositionSearchResponse

Name	Type / Size	Description
AssgNo.	String	This displays the assignement number.
Job Code	String	This displays the job code.
JobLong Description	String	This displays the job long description.

GetBlindSearch Method

Use this service method to search for position definitions records based upon requested search criteria.
WcfHR.HRPositionSpecificServices.GetBlindpositionSearchMethod (WcfHR. BlindPositionSearchRequest)
Records: Request (input) – Blind Position search request
Returns: Returns Blind Position Search response. **Data Contract – BlindPositionSearchResponse**

Name	Type / Size	Description
AssgLongDesc	String	This displays the assignment long description.
AssgNo	String	This displays the assignment number.
AssgShortDes	String	This displays the assignment short description.
BudgStat	String	This displays the budget status.
JobCode	String	This displays the job code.
JobLongDesc	String	This displays the job long description.
JobShortDesc	String	This displays the job short description.
Location	String	This displays the location.
LocLongDesc	String	This displays the location long description.
LocShortDesc	String	This displays the location short description.
Pcn	String	This displays the pcn.

Name	Type / Size	Description
PcnLongDesc	String	This displays the position long description.
PcnShortDesc	String	This displays the position short description.
PcnPosLong	String	This displays the pcn position long.
PcnUnique Key	String	This displays the pcn unique key. This is derived from the table HRPcnTble.
Position	String	This displays the position.

Data Contract – BlindPositionSearchRequest

Name	Type / Size	Description
Effective Date	String	Effective date is used to add selection criteria to confirm that the effective date is between effective begin and end dates.
EntityID	String	This displays the entity ID.
Search Assignment	String	This displays the search assignment.
SearchJob	String	This displays the search job.
SearchLocation	String	This displays the search location.
SearchPosition	String	This displays the search position.
Search String	String	This displays the search string.

GetBlindPositionSearchByCrieriaMethod

Use this service method to search for position definitions records based upon requested search criteria.
WcfHR.HRPositionSpecificServices.GetBlindpositionSearchByCriteriaMethod (WcfHR. BlindPositionSearchByCriteriaRequest)
Records: Request (input) – Blind Position search request
Returns: Returns Blind Position Search response. **Data Contract – BlindPositionSearchResponse**

Name	Type / Size	Description
AssgLongDesc	String	This displays the assignment long description.
AssgNo	String	This displays the assignment number.
AssgShortDes	String	This displays the assignment short description.
BudgStat	String	This displays the budget status.

Name	Type / Size	Description
JobCode	String	This displays the job code.
JobLongDesc	String	This displays the job long description.
JobShortDesc	String	This displays the job short description.
Location	String	This displays the location.
LocLongDesc	String	This displays the location long description.
LocShortDesc	String	This displays the location short description.
Pcn	String	This displays the pcn.
PcnLongDesc	String	This displays the position long description.
PcnShortDesc	String	This displays the position short description.
PcnPosLong	String	This displays the pcn position long.
PcnUnique Key	String	This displays the pcn unique key. This is derived from the table HRPcnTble.
Position	String	This displays the position.

Data Contract – BlindPositionSearchRequest

Name	Type / Size	Description
Effective Date	String	Effective date is used to add selection criteria to confirm that the effective date is between effective begin and end dates.
EntityID	String	This displays the entity ID.
Search Assignment	String	This displays the search assignment.
SearchJob	String	This displays the search job.
SearchLocation	String	This displays the search location.
SearchPosition	String	This displays the search position.
Search String	String	This displays the search string.

GetEmployeePositionSalariesMethod

Use this service method to get employee position salaries.

WcfHR.HRPositionSpecificServices.GetEmployeePositionSalariesMethod (WcfHR. EmployeePositionSalariesRequest)
Records: Request (input) – Employee Position Salaries request
Returns: Employee Position Salaries response.

Data Contract – EmployeePositionSalaryResponse

Name	Type / Size	Description
Amount	String	This displays the assignment long description.
Axp	String	This displays the assignment number.
PayEffBeg	DateTime	This displays the assignment short description.
PayEffEnd	DateTime	This displays the budget status.
Range	String	This displays the range.
SchedType	String	This displays the schedule type.
Schedule	String	This displays the schedule.
Step	String	This displays the step.

Data Contract – EmployeePositionSalaryRequest

Name	Type / Size	Description
Begin Date	DateTime	This displays the start date.
Effective Date	DateTime	Effective Date is used to add selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
EndDate	DateTime	This displays the end date.
EntityID	String	This displays the entity ID.
Range	String	This displays the range.
Schedule	String	This displays the schedule.

GetEmployeePositionTitleMethod

Use this service method to get employee position title.
WcfHR.HRPositionSpecificServices.GetEmployeePositionTitleMethod (WcfHR. EmployeePositionTitleRequest)
Records: Request (input) – Employee Position Title request
Returns: Employee Position Title response.

Data Contract – EmployeePositionTitleResponse

Name	Type / Size	Description
ActAnn	String	This displays the actual annual value.
ActHrly	String	This displays the actual hourly value.
BargLongDesc	DateTime	This displays the bargaining long description.
EmployeeID	DateTime	This displays the employee ID.
Entity ID	String	This displays the entity ID.
IndxKey	string	This displays the index key.
PayBeg	DateTime	This displays the pay beginning date.
PayEnd	DateTime	This displays the pay end date.
PayStat	String	This displays the pay status.
PayUniqueKey	String	This displays the pay unique key. This value is derived from HREmpPay table.
PcnUniqueKey	String	This displays the Pcn unique key. This value is derived from HRPcn table.
Position	String	This displays the position.
PosLongDesc	String	This displays the position long description.
ReCalc	String	This displays the recalculation.
RecTye	String	This displays the record type.
SuperID	String	This displays the supervisor ID.

Data Contract – Employee Position Title Request

Name	Type / Size	Description
EntityID	String	This displays the entity ID.
EmployeeID	String	This displays the employee ID.

GetExtEmpPayMethod

Use this service method to get extension employee pay positions.
WcfHR.HRPositionSpecificServices.GetExtEmpPayPositionsMethod (WcfHR.
ExtEmppayPositionRequest)
Records: Request (input) – Position Change Request
Returns: Position Change response.

Data Contract – ExtEmpPayPositionResponse

Name	Type / Size	Description
AssgNo	String	This displays the assignment number.
Calendar	String	This displays the calendar.
CldDs	String	This displays the calendar description.
Department		This displays the department.
EmpUnique Key	String	This displays the employee unique key. This is derived from the HREmpmStrTable.
JobCode	String	This displays the job code.
JobLongDesc	String	This displays the job long description.
JobShortDesc	String	This displays the job short description.
Location	String	This displays the location.
LocLongDesc	String	This displays the location long description.
LocShortDesc	String	This displays the location short description.
Pay UniqueKey		This displays the pay unique key. This is derived from HREmppaytable.
Pcnfte	String	This displays the pcnfte.
PcnLongDesc	String	This displays the position long description.
PcnShortDesc	String	This displays the position short description.
PcnPosLong	String	This displays the pcn position long.
PcnUnique Key	String	This displays the pcn unique key. This is derived from the table HRPcnTble.
Position	String	This displays the position.
SuperID	String	This displays the supervisor ID.
Type	String	This displays the type.

GetPositionData

This service returns information about PCNs.

Name	Type	Description
	String	If a PCN number is supplied, only information for that position will be returned.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri =
"http://localhost/FusionOSServices/v1/ONESolution/HumanResources/Positions/HOUR
S?fields=cdhno,cdhtitle,longdesc,pcn,cdhrelcd06+TCPType&cdhrelcd05=TCP&pcn=0105
+like";


    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        while (uri != null)
        {
            string result = wc.DownloadString(uri);

            var response = JObject.Parse(result);

            var count = (string)response["RequestResponse"]["Count"];
            var items =
(JContainer)response["RequestResponse"]["DATA"]["POSITION"];

            foreach (var item in items )
            {
                foreach (var hour in item["HOURS"])
                {
                // Your code goes here
                }
            }
            // responses with more then 1000 positions are broken up into
multiple requests
            uri = (string)response["RequestResponse"]["NextUri"];
        }
    }
}
```

Sample Response: JSON

```
{
    "RequestResponse": {
        "Count": "3",
        "DATA": {
            "POSITION": [
                {
                    "LONGDESC": "CITY COUNCIL",
                    "PCN": "01050000",
                    "HOURS": [
                        {}
                    ]
                },
                {
                    "LONGDESC": "CITY COUNCIL",
                    "PCN": "01050000",
                    "HOURS": [
                        {
                            "HOUR": [
                                {
                                    "CDHNO": "3002",
                                    "CDHTITLE": "SALARY",
                                    "TCPTYPE": "REG"
                                },
                                {
                                    "CDHNO": "3402",
                                    "CDHTITLE": "VACATION USED",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3422",
                                    "CDHTITLE": "SICK USED",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3432",
                                    "CDHTITLE": "SICK USED-FAMILY",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3442",
                                    "CDHTITLE": "PERSONAL USED",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3462",
                                    "CDHTITLE": "ADMIN USED",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3488",
                                    "CDHTITLE": "COMP USED",
                                    "TCPTYPE": "LEAVE"
                                }
                            ]
                        }
                    ]
                },
                {
                    "LONGDESC": "CITY COUNCIL",
                    "PCN": "01050000",
                    "HOURS": [
                        {
                            "HOUR": [
                                {
                                    "CDHNO": "3002",
                                    "CDHTITLE": "SALARY",
                                    "TCPTYPE": "REG"
                                },
                                {
                                    "CDHNO": "3402",
                                    "CDHTITLE": "VACATION USED",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3422",
                                    "CDHTITLE": "SICK USED",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3432",
                                    "CDHTITLE": "SICK USED-FAMILY",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3442",
                                    "CDHTITLE": "PERSONAL USED",
                                    "TCPTYPE": "LEAVE"
                                },
                                {
                                    "CDHNO": "3462",
                                    "CDHTITLE": "ADMIN USED",
                                    "TCPTYPE": "LEAVE"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    }

}
```

Query string options:

SkipRows: The number of rows at the begining of the query to not report.

TakeRows: The number of rows from the query to report. If left blank the default is 1000.

?TakeRows=10 results in only 10 rows being returned.

?SkipRows=10 results in all rows after row 10 being returned.

?SkipRows=10&TakeRows=10 results in rows 11 through 20 being returned.

Fields: A list of table columns to be included in the query. If this is included in the query string then only the listed columns will be reported.

(note that the examples below are for HREmpmstr, but similar ideas can be applied to any table)

?fields=fname,lname results in only the first and last name being reported.

Columns can be renamed by appending the new name after the default name.

?fields=fname+FirstName,lname+LastName

Filters:

Any column name can be added to the query string as a filter. For example

?fname=Dan

?hdt=1/1/2019

By default the filter comparison is equality. Other comparisons are available and can be appended to the filter.

"?fname=Dan+like " will select Dan, Daniel and Dannie

"?hdt=1/1/2019+gt" will include all dates after 1/1/2019

Available comparisons are:

"ge", greater than or equal to (>=)

"gt", greater than (>)

"isnotnull", is not null

"isnull", is null

"le", less than or equal to (<=)

"like", like

"lt", less than (<)

"ne", not equal (!= , <>)

"notlike", not like

GetPositionHoursData

This service returns information about PCN and the Hours associated with their pay classes.

Name	Type	Description
	String	If a PCN number is supplied, only information for that position will be returned.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri =
"http://localhost/FusionOSServices/v1/ONESolution/HumanResources/Positions/HOUR
S?fields=cdhno,cdhtitle,longdesc,pcn,cdhrelcd06+TCPType&cdhrelcd05=TCP";


    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");
        while (uri != null)
        {
            string result = wc.DownloadString(uri);

            var response = JObject.Parse(result);

            var count = (string)response["RequestResponse"]["Count"];
            var items =
(JContainer)response["RequestResponse"]["DATA"]["POSITION"];

            foreach (var item in items )
            {
                // Your code goes here
            }
            // responses with more then 1000 positions are broken up into
multiple requests
            uri = (string)response["RequestResponse"]["NextUri"];
        }
    }
}
```

Sample Response: JSON

```
{
    "RequestResponse": {
        "Count": "5",
        "DATA": {
            "POSITION": [
                {
                    "LONGDESC": "MANAGEMENT ANALYST",
                    "PCN": "02200000"
                },
                {
                    "LONGDESC": "ASSISTANT CITY CLERK",
                    "PCN": "02202000"
                },
                {
                    "LONGDESC": "CITY CLERK",
                    "PCN": "02204000"
                },
                {
                    "LONGDESC": "RECORDS MANAGER",
                    "PCN": "02206000"
                },
                {
                    "LONGDESC": "HUMAN RESOURCES MANAGER",
                    "PCN": "02208000"
                }
            ]
        }
    }
}
```

Query string options:

SkipRows: The number of rows at the beginning of the query to not report.
TakeRows: The number of rows from the query to report. If left blank the default is 1000.

?TakeRows=10 results in only 10 rows being returned.

?SkipRows=10 results in all rows after row 10 being returned.
?SkipRows=10&TakeRows=10 results in rows 11 through 20 being returned.

Fields: A list of table columns to be included in the query. If this is included in the query string then only the listed columns will be reported.

(note that the examples below are for HREmpmstr, but similar ideas can be applied to any table)

?fields=fname,lname results in only the first and last name being reported.
Columns can be renamed by appending the new name after the default name.
?fields=fname+FirstName,lname+LastName

Filters:

Any column name can be added to the query string as a filter. For example

?fname=Dan
?hdt=1/1/2019

By default the filter comparison is equality. Other comparisons are available and can be appended to the filter

"?fname=Dan+like " will select Dan, Daniel and Dannie
"?hdt=1/1/2019+gt" will include all dates after 1/1/2019

Available comparisons are:

"ge", greater than or equal to (>=)
"gt", greater than (>)
"isnotnull", is not null
"isnull", is null
"le", less than or equal to (<=)
"like", like
"lt", less than (<)
"ne", not equal (!= , <>)
"notlike", not like

GetPositionHoursData

Use this service method to get the position.
WcfHR.HRPositionSpecificServices.GetPositionsMethod (WcfHR.PositionRequest)
Records: Request (input) – Position Request Returns: Position Response.

Data Contract – PositionResponse

Name	Type / Size	Description
AssgLongDesc	String	This displays the assignment long description.
AssgNo	String	This displays the assignment number.
AssgShortDes	String	This displays the assignment short description.
BudgStat	String	This displays the budget status.
JobCode	String	This displays the job code.
JobLongDesc	String	This displays the job long description.
JobShortDesc	String	This displays the job short description.
Location	String	This displays the location.
LocLongDesc	String	This displays the location long description.
LocShortDesc	String	This displays the location short description.

Pcn	String	This displays the pcn.
PcnLongDesc	String	This displays the position long description.
PcnShortDesc	String	This displays the position short description.
PcnPosLong	String	This displays the pcn position long.
PcnUniqueKey	String	This displays the pcn unique key. This is derived from the table HRPcnTble.
Position	String	This displays the position.
Range	String	This displays the range.
Schedule	String	This displays the schedule.
Type	String	This displays the type

Name	Type / Size	Description
SuperID	String	This displays the supervisor ID.

Data Contract – Position Request

Name	Type / Size	Description
EntityID	String	This displays the entity ID.
PositionID	String	This displays the employee ID.
Pcn	String	This displays the pcn.

GetPositionChangeMethod

Use this service method to get the change position.
WcfHR.HRPositionSpecificServices.GetPositionsChangeMethod (WcfHR.PositionChangeRequest) Records: Request (input) – Position Change Request Returns: Position Change Response.

Data Contract – PositionChangeResponse

Name	Type / Size	Description
AssgNoLongDesc	String	This displays the assignment number long description.
AssgNo	String	This displays the assignment number.
AssgStatLongDesc	String	This displays the assignment status long description.
AssgStat	String	This displays the assignment status.
BargLongDesc	String	This displays the bargaining unit long description. The value is derived from HRBargTble table.
BargUnit	String	This displays the bargaining unit.
Calendar	String	This displays the calendar.
CldDs	String	This displays the calendar description.
Department	String	This displays the department.
DeptCodeVal	String	This displays the department code value.
DeptLongDesc	String	This displays the department long description.
EmpCodeVal	String	This displays the employee code value.
EmpLongDesc	String	This displays the employee long description.

Name	Type / Size	Description
EmpUniqueKey	String	This displays the employee unique key.
JobCode	String	This displays the job code.
JobLongDesc	String	This displays the job long description.
JobShortDesc	String	This displays the job short description.
Location	String	This displays the location.
LocLongDesc	String	This displays the location long description.
LocShortDesc	String	This displays the location short description.
Pcn	String	This displays the pcn.
PcnEffort		This displays the pcn effort.
Pcnfte		This displays the Pcnfte.
PcnLongDesc	String	This displays the position long description.
PcnShortDesc	String	This displays the position short description.
PcnPosLong	String	This displays the pcn position long.
PcnUniqueKey	String	This displays the pcn unique key. This is derived from the table HRPcnTble.
Position	String	This displays the position.
RecType	String	This displays the record type.
SuperID	String	This displays the supervisor ID
TotFte		This displays the total fte.
Type	String	This displays the type.

Data Contract – Position Change Request

Name	Type / Size	Description
Effective Date	DateTime	Effective Date When used, adds selection criteria to confirm that the Effective Date is between Effective Begin and End Dates.
EmployeeID	String	This displays the employee ID.
Entity ID	String	This displays the pcn.

GetPositionEarningDistributionMethod

Use this service method to get the position earning distribution method.
WcfHR.HRPositionSpecificServices.GetPositionsEarningDistributionMethod
(WcfHR.PositionEarningDistributionRequest)
Records: Request (input) – Position Earning Distribution Request
Returns: Position Earning Distribution Response.

Data Contract – Position EarningDistributionResponse

Name	Type / Size	Description
Earn unique key	DateTime	This displays the earn unique key. This is derived from the HREarnDist table.
GL key	String	This displays the GL key.
GLKeyDesc	String	This displays key description.
GLLedger	String	This displays the GL ledger.
GLLedgerDesc	String	This displaysGL description.
GLObject	String	This displays the GL object.
Jlkey	String	This displays the Jl key.
Jlkey Desc	String	This displays the Jl key description.
JlLedger	String	This displays the Jl ledger.
JLLedger Desc	String	This displays the Jl ledger description.
JlObject	String	This displays the Jl object.
Pay Unique Key	String	This displays the pay unique key. This is derived from HR Emppay table.
Percent	String	This displays the percent.

Data Contract – Position Distribution Earning Request

Name	Type / Size	Description
Effective Date	DateTime	This displays the effective date, it adds selection criteria to confirm that the effective date is between effective begin and end dates.
EmployeeID	String	This displays the employee ID.
Position	String	This displays the position.

GetPositionSearchMethod

Use this service method to get the position search.
WcfHR.HRPositionSpecificServices.GetPositionSearchMethod (WcfHR.GetPositionSearch)
Records: Request (input) – Position Search Request Returns: Position Search Response.

Data Contract – PositionSearchResponse

Name	Type / Size	Description
AssgNo	String	This displays the assignment number.
JobCode	String	This displays the job code.
JobLongDesc	String	This displays the job long description.
JobShortDesc	String	This displays the job short description.
Location	String	This displays the location.
LocLongDesc	String	This displays the location long description.
LocShortDesc	String	This displays the location short description.
PcnLongDesc	String	This displays the position long description.
PcnShortDesc	String	This displays the position short description.
PcnPosLong	String	This displays the pcn position long.
Position	String	This displays the position.

Data Contract – Position Request

Name	Type / Size	Description
EntityID	String	This displays the entity ID.
PositionID	String	This displays the employee ID.

Human Resources – HRRegulation of pcn Services

GetBudgetedAmountPerPcnMethod

Use this service method to get budgeted amount per pcn.
WcfHR.HR RegulationofPcnServices.GetBudgetedAmountPerPcnMethod (WcfHR. BudgetedSumperPcnRequest)
Records: Request (input) – Budgeted sum per pcn request
Returns: Returns

Data Contract – BudgetedSumperPcnRequest

Name	Type / Size	Description
EffectiveDate	DateTime	This displays the effective date, it adds selection criteria to confirm that the effective date is between effective begin and end dates.
Pcn	String	This displays the pcn.

GetBudgetedFtePerPcnMethod

Use this service method to get budgeted fte per pcn.
WcfHR.HR RegulationofPcnServices.GetBudgetedFtePerPcnMethod (WcfHR. BudgetedSumperPerPcnRequest)
Records: Request (input) – Budgeted sum per pcn request
Returns: Returns

Data Contract – BudgetedSumperPerPcnRequest

Name	Type / Size	Description
EffectiveDate	DateTime	This displays the effective date, it adds selection criteria to confirm that the effective date is between effective begin and end dates.
Pcn	String	This displays the pcn.

Nucelus – CommonCodeServiceClass

This is the primary class containing the web service entry points for working with Common Codes from the Finance Enterprise module.

GetCommonCodesMethods

WcfNU.CommonCodeServices.GetCommonCodesMethods
This method gets an array of common codes from the Finance Enterprise module.

Parameters

records: (input) Ledger, code category. The ledger is optional. If no ledger is specified then the "@@", or wildcard ledger will be used. Specifies which common code group to select. The Code Category parameter is required.

returns:

LookUpCommonCodes Methods

WcfNU.CommonCodesService.LookUpCommonCodes Method (WcfNu.LookUpCommonCodesRequest)

Parameters

record: (input) request. The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of CdCodesMstrSubset records based upon the search criteria, otherwise returns null.

Data Contract - LookUpCommonCodesRequest

Name	Type / Size	Description
MaxRows	String	Maximum number of rows to return No limit on rows when null or undefined.
OrderByFields	String	Order By Parameter Field
Parameters	String	Array of Codes Lookup Parameters
SelectDistinct	String	Select Distinct
SelectFields	String	Select Fields

Nucleus – License Service Class

Provides services to query the license.

IsApplicationLicensedMethod

Use this service to see if the application is licensed.
WcfNU.LicenseServiceClass.IsApplicationLicensedMethod (WcfHR. BudgetedSumperPcnRequest) Records: Request (input) – The application to check. Returns: True if the application is licensed, otherwise, false.

IsSubsystemLicensedMethod

Use this service to check to see if the subsystem is licensed.
WcfNU.LicenseServiceClass.IsSubsystemLicensedMethod
Records: Request (input) – The subsystem to check. Returns:
True if the subsystem is licensed, otherwise, false.

Nucleus – User Service Class

Contains the services for reading and writing user data.

CanYouMethod

Use this service to check to see if the user can run the specified masks.
WcfNU.UserServiceClass.CanYouMethod Records: Request (input) – The masks to check. Returns: The results for each of the masks sent in.

CanYouRunMaskMethod

Use this service to check to see if the user can run the specified mask.
WcfNU.UserServiceClass.CanYouRunMaskMethod Records:
Request (input) – The mask to check for access to. Returns: True if the user has access, otherwise, false.

CanYouRunMasksMethod

Use this service to check to see if the user can run the specified masks.
WcfNU.UserServiceClass.CanYouRunMasksMethod Records:
Request (input) – The masks to check for access to. Returns: The
results for each of the masks sent in.

GetAllUsersMethod

Use this service to get all the users summary information from the us_usno_mstr table.

WcfNU.UserServiceClass.GetAllUsersMethod
Returns: The user summary data.
Data Contract – UserDataSummary

Name	Type / Size	Description
Description	String	Gets or sets the description of the user.
E-mail	String	Gets or sets the email address of the user.
IsRegistered	String	Gets or sets if the user is already registered with SPSOne.
UserID	String	Gets or sets the user id.
UserName	String	Gets or sets the user name.

GetAllUsersCompressedMethod

Use this service to get all the users summary information from the us_usno_mstr table.
WcfNU.UserServiceClass.GetAllUsersCompressedMethod
Returns: The user summary data in a compressed blob.

GetUserDataMethod

Use this service to fetch the user data.
WcfNU.UserServiceClass.GetUserDataMethod
Input: UserSID. The SID of the user for whom you want the data.
Returns: The user data.

QuickDebugFlushMethod

Use this method to flush the security for the specified user.
WcfNU.UserServiceClass.QUickDebugFlushMethod
Input: UserID. The UserID to flush.

RegisterExistingUserMethod

Use this method to register the existing user.
WcfNU.UserServiceClass.RegisterExistingUserMethod
Input: UserID, UserSID, user name. The userID and SID of the user. The user name used by spsone for the user.

RegisterExistingUserMethod

Use this method to register the existing user.
WcfNU.UserServiceClass.RegisterExistingUserMethod
Input: UserID, UserSID, user name. The userID and SID of the user. The user name used by spsone for the user.

SaveUserDataMethod

Use this method to register the existing user.
WcfNU.UserServiceClass.SaveUserDataMethod
Input: UserID, UserSID, user name. The userID and SID of the user. The user name used by spsone for the user.

Person Entity

You use this method to store inventory commodity services.

LookUpCommodityMethod

WcfPE.PECommodityServices.LookUpCommodityMethod (WcfPE.LookUpCommodityRequest)
You use this method to lookup Stores Inventory Commodity records based upon requested search criteria.

Parameters

records: (input) Request - The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of PeCommMstr records when Commodity ID's meets the search criteria, otherwise returns null.

ValidateCommodityMethod

WcfPE.PECommodityServices.ValidateCommodityMethod (WcfPE.ValidateCommodityRequest)
You use this method to validate a Stores Inventory Commodity ID in the PeCommMstr table.

Parameters
records: (input) Request - Enter a valid Commodity ID.
returns: Returns subset of PeCommMstr record when the Commodity ID is found, otherwise returns null.

Person Entity – Vendor Service Class

Contains the services for Person/Entity Vendor Services.

InsertVendorMethod

Use this service to check to see if the user can run the specified masks.
WcfPE.VendorServiceClass.InsertVendorMethod
Records: InsertRequest (input) – The details of the Person\Entity Vendor to be inserted, along with a boolean parameter indicating whether to perform rudimentary validations or not.
Returns: Returns null on failed Insert attempt, otherwise response will contain PeId created. Details about the error encountered can be found in the Output Context Data.

Data Contract - WcfPE.VendorIntrestRequest

Name	Type / Size	Description
Record		The PEID of the vendor for the AP record.
Validate		Flag to denote whether to perform rudimentary validations or not

LookUpVendorMethod

Use this service to lookup Person\Entity Vendor records based upon requested search criteria.
WcfPE.VendorServiceClass.LookUpVendorMethod
Records: Request (input) – The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.
Returns: Returns an array of PeNameMstr records when Vendor ID's meets the search criteria, otherwise returns null.

ValidateVendorMethod

Use this service to validate a Person\Entity Vendor ID.
WcfPE.VendorServiceClass.ValidateVendorMethod Records:
Request (input) – Input, the Vendor ID to valid.
Returns: Returns subset of PeNameMstr, PeAddrDtl, PeEmailDtl and PePhoneDtl records when the Vendor ID is found, otherwise returns null.

Purchase Order

You use this to store inventory services.

LookUpPODetailsMethod

WcfPO.POServices.LookUpPODetialsMethods (WcfPO.LookUpPODetialsRequest)
You use this method to lookup Stores Purchase Order records based upon requested search criteria.
Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of PoDtl records when the search criteria is met, otherwise returns null.

LookUpPOLineItemDetailsMethod

WcfPO.POServices.LookUpPOLineItemDetailsMethod (WcfPO.LookUpPOLineItemDetialsRequest)
You use this method to lookup Stores Purchase Order records based upon requested search criteria.
Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of PoDtl records when the search criteria is met, else returns null.

Payroll

GetLeaveDefs

Use this service to return the types of leave possible and the hours that affect those leaves.

Sample Code C#:

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri =
"http://localhost/FusionOSServices/v1/ONESolution/Payroll/LeaveDefinitions";


    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        while (uri != null)
        {
            string result = wc.DownloadString(uri);

            var response = JObject.Parse(result);
            var transactions =
(JContainer)response["RequestResponse"]["DATA"]["Leaves"];

            foreach (var transaction in transactions )
            {
                // Your code goes here
                string leaveID = (string)transactions["LEAVEID"];
                string leaveDescription = (string)transactions["Title"];
                var hours = (JContainer)transaction["CDHS"];
                foreach(var hour in hours)
                {
                    string hourNumber = hour["CDH"];
                    string hourTitle = hour["Title"];
                    string effect = hour["Effect"];
                }
            }
            // responses with more then 1000 positions are broken up into
multiple requests
            uri = (string)response["RequestResponse"]["NextUri"];
        }
    }
}
```

Sample Responses: JSON

```
            {
                "RequestResponse": {
                "Count": "none",
                 "DATA": {
                   "LEAVES": [
                     {
                       "LEAVEID": "7",
                       "Title": "Vacation Balance",
                       "CDHS": [
                         {
                           "CDH": "3400",
                           "Title": "VACATION ACCRUAL",
                           "Effect": "+"
                         },
                         {
                           "CDH": "3402",
                           "Title": "VACATION USED",
                           "Effect": "-"
                         },
                         {
                           "CDH": "3404",
                           "Title": "VACATION ADJUSTMENT",
                           "Effect": "+"
                         },
                         {
                           "CDH": "3406",
                           "Title": "VACATION PAYOUT",
                           "Effect": "-"
                         }
                   ]
                 }
               }
             }
```

GetLeaveTransactions Method

Use this REST service to get information about leave transactions. Each transaction will contain the name of the leave, the type of transaction and the number of hours. Additional information can be request via parameters.

Name	Type / Size	Description
	String	If supplied, the response will be limited to just those transactions associated with the given employee.

Allowed values for the Fields paramater include: hrpeid (employee ID), pyppercc (pay period number), checkno (pay check or EFT statement number), cdhno (the hour number), amount (the dollar amount associated with the transaction) and numcd (the payroll num cd).

These values can also be use as filters, for example, "?trnsdate=1/1/2019" would limit the response to just those transaction occuring on January 1st, 2019.

Query string options

SkipRows: The number of rows at the beginning of the query to not report.

TakeRows: The number of rows from the query to report. If left blank the default is 1000.

?TakeRows=10 results in only 10 rows being returned.

?SkipRows=10 results in all rows after row 10 being returned.

?SkipRows=10&TakeRows=10 results in rows 11 through 20 being returned.

Fields: A list of table columns to be included in the query. If this is included in the query string then only the listed columns will be reported.

(note that the examples below are for HREmpmstr, but similar ideas can be applied to any table)

?fields=fname,lname results in only the first and last name being reported.

Columns can be renamed by appending the new name after the default name.

?fields=fname+FirstName,lname+LastName

Filters:

Any column name can be added to the query string as a filter. For example:

?fname=Dan

?hdt=1/1/2019

By default, the filter comparison is equality. Other comparisons are available and can be appended to the filter.

"?fname=Dan+like " will select Dan, Daniel and Dannie

"?hdt=1/1/2019+gt" will include all dates after 1/1/2019

Available comparisons are:

"ge", greater than or equal to (>=)

"gt", greater than (>)

"isnotnull", is not null

"isnull", is null

"le", less than or equal to (<=)

"like", like

"lt", less than (<)

"ne", not equal (!= , <>)

"notlike", not like

GetTimecards Method

Use this REST service to return Timecard data.

Allowed values for the Fields paramater include: empid (employee ID), trandt (Timecard Date), hrs (Timecard Hours), cdh (the hour number) and rectype (the payroll record type).

these values can also be use as filters, for example, "?trandt=1/1/2019" would limit the response to just those transaction occurring on January 1st, 2019.

Query string options:

SkipRows: The number of rows at the beginning of the query to not report.

TakeRows: The number of rows from the query to report. If left blank the default is 1000.

?TakeRows=10 results in only 10 rows being returned.

?SkipRows=10 results in all rows after row 10 being returned.

?SkipRows=10&TakeRows=10 results in rows 11 through 20 being returned.

Fields: A list of table columns to be included in the query. If this is included in the query string, then only the listed columns will be reported.

(note that the examples below are for HREmpmstr, but similar ideas can be applied to any table)

?fields=empid,hrs results in only the employee id and hours being reported.

Columns can be renamed by appending the new name after the default name.

?fields=empid+EmployeeID,trandt+TimecardDate

Filters:

Any column name can be added to the query string as a filter. For example:

?empid=E0001

?trandt=1/1/2019

By default, the filter comparison is equality. Other comparisons are available and can be appended to the filter

"?fname=Dan+like " will select Dan, Daniel and Dannie

"?hdt=1/1/2019+gt" will include all dates after 1/1/2019

Available comparisons are:

"ge", greater than or equal to (>=)

"gt", greater than (>)

"isnotnull", is not null

"isnull", is null

"le", less than or equal to (<=)

"like", like

"lt", less than (<)

"ne", not equal (!= , <>)

"notlike", not like

PutTimecards Method

Use this REST service to insert Timecard data.

Expected input is a JSON array of objects, with each object containing one timecard entry.

Allowed elements in each timecard entry are EmpId, TranDt, Cdh, Hrs, BatchId, PerCc, Sort, Group, Misc1, Misc2, Misc3, Misc4, Misc5, BegDt, RecType, NumCd, RingIn, RingOut, Rate, Amt, RateCd1, RateCd2, RateCd3, RateCd4, RateCd5, ReasonCd, OvrHrs, OvrRt, OvrAmt, OvrNum, Status, JobNo, Orig, Text, Link, EmpId2, Account, Wo, PayClass, Pos, Step, GlGr, GlKey, GlObj, JlGr, JlKey, JlObj, User1, User2, User3, User4, User5, User6, User7, User8, User9, AprvCd1, AprvCd2, AprvCd3, AprvCd4, AprvCd5, TranOp, SystemId, ProcessedDt, MiscDesc, WfCode, Eqno, Ifpy980Flag.

While all of these elements are optional, the first four, EmpId, TranDt, Cdh and Hrs are recommended.

Elements included in the JSON but not in this list will be ignored.

Hour values with more than 5 decimal places of precision will be accepted, but truncated. e.g. 5.123456 will be accepted as 5.12345

The response from this call will include the number of inserted records and a list of errors, if any. Any error will result in the entire submission being rejected and the count reported will be zero.

LookUprdTimeCardDetailsMethod

WcfPY.PYTimeCardServices.LookUpTimeCardDetialsMethods (WcfPY.LookUpTimeCardRequest)
You use this method to lookup Payroll TimeCard Detail records based upon requested search criteria.

Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of TimeCard detail records based upon the search criteria, otherwise returns null.

Stores Inventory- SIInventory

LookUpInventoryMethod

WcfSI.SIInventoryService.LookUpInventorymethod (WcfSI.LookUpInventoryRequest)
You use this method to lookup Stores Inventory records based upon requested search criteria.

Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of SiiInvtryDtl records when Product ID's and Warehouse's meet the search criteria, otherwise returns null.

ValidateInventoryMethod

WcfSI.SIInventoryService.ValidateInventorymethod (WcfSI.ValidateInventoryRequest)
You use this method to validate a Stores Inventory Product ID and Warehouse in the SiiInvtryDtl table.

Parameters

records:	(input) Input, the Product ID and Warehouse to validate.
returns:	Returns subset of SiiInvtryDtl record when the Product ID and Warehouse are found, otherwise returns null.

Stroes Inventory- SIOrder

InsertandValidateOrderRecordMethod

WcfSI.SIOrderServrice.InsertandValidateOrderRecordmethod (WcfSI.InsertOrderRequest)
You use this method to insert a single Stores Inventory Order record with rule validation.

Parameters

records: (input) Insert Order Request.

returns: Valid Insert Order Response if no errors, otherwise null.

Data Contract - WcfSI.InsertOrderResponse

Name	Type / Size	Description
LineNum	String	This displays the line number.
OrderID	String	This displays the Order ID.

Data Contract - WcfSI.InsertOrderRequest

Name	Type / Size	Description
AcctAmount	Decimal	This displays the amount.
AcctGlGr	String	This displays the GL ledger.
AcctGLKey	String	This displays the GL key.
AcctGLObj	String	This displays the GL object.
AcctJLGr	String	This displays the Jl Ledger
AcctJlKey	String	This displays the JL key.
AcctJLObj	String	This displays the JL object.
AcctPercent	String	This displays the percentage.
AcctWo	String	This displays the work order.
ItemLinenum	String	This displays the line number.
ItemlocOrder	String	This displays the location sort order.
ItemPRCd	String	This displays the price code.
ItemProd ID	String	This displays the product ID.

Name	Type / Size	Description
ItemQntyOrd	String	This displays the quantity ordered.
ItemSeqNo	String	This displays the sequence number.
ItemSimCode	String	This displays the line status.
ItemUnits	String	This displays the units.
ItemWhse	String	This displays the warehouse.
OderAprv	String	This displays the approved by.
OrderAprvDt	String	This displays the approval date.
OrderContact	String	This displays the order contact.
Order Desc	String	This displays the order description.
OrderEndUse	String	This displays the order end use.
OrderID	String	This displays the order ID.
OrderMisc	String	This displays the miscellaneous code.
OrderOrdCd 01 - 10	String	This displays the order code 01 - 10.
OrderPEAddr	String	This displays the address code.
OrderPEID	String	This displays the Customer ID.
OrderPEPO	String	This displays the Customer PO.
OrderPEType	String	This displays the transaction code.
OrderPrCd	String	This displays the price code.
OrderPrepID	String	This displays the prepare ID.
OrderRgrDt	DateTime	This displays the required date.
OrderRgsBy	String	This displays the requested by.
OrderRgsDt	String	This displays the requested date.
OrderSecCode	String	This displays the user security code.

Name	**Type / Size**	**Description**
Order Status	String	This displays the order status.
OrderTrnFmt	String	This displays the transaction format.
OrderWhse1	String	This displays the primary warehouse.
OrderWhse2	String	This displays the secondary warehouse.
OrderYearID	String	This displays the year ID.
Seed Code	String	This displays the seed code.
User name	String	This displays the user name.

InsertOrderRecordsMethod

WcfSI.SIOrderSerrvice.InsertOrderRecordsmethod (WcfSI.InsertOrderRequest)
You use this method to inserts one or more Stores Inventory Order records without rule validation. Removing validation from the process allows for better performance when dealing with high volumes of transactions.

Parameters

records:	(input) Insert Order Request array.
returns:	Valid Insert Order Response if no errors, otherwise null.

Data Contract - WcfSI.InsertOrderResponse

Name	Type / Size	Description
LineNum	String	This displays the line number.
OrderID	String	This displays the Order ID.

LookupAllocatedOrdersMethod

WcfSI.SIOrderSerrvice.LookupAllocatedOrdersmethod (WcfSI.LookupOrderRequest)
You use this method to lookup Allocated Orders based upon requested search criteria.

Parameters

records:	(input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.
returns:	Returns an array of Allocated Orders when search criteria met, otherwise returns null.

LookupBackOrdersMethod

WcfSI.SIOrderSerrvice.LookupBackOrdersMethod (WcfSI.LookupOrderRequest)
You use this method to lookup Allocated Orders based upon requested search criteria.

Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of Back Orders when WO ID's meets the search criteria, otherwise returns null.

LookupIssuedOrdersMethod

WcfSI.SIOrderSerrvice.LookupIssuedOrdersMethod (WcfSI.LookupOrderRequest)
You use this method to lookup Allocated Orders based upon requested search criteria.

Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of Allocated Orders when search criteria met, otherwise returns null.

LookupOrdersMethod

WcfSI.SIOrderSerrvice.LookupOrdersMethod (WcfSI.LookupOrderrequest)
You use this method to lookup Order details based upon requested search criteria.

Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of Allocated Orders when search criteria met, otherwise returns null.

UpdateOrderRecordMethod

WcfSI.SIOrderService.UpdateOrderRecordMethod(WcfSI.UpdateOrderRequest)
You use this method to lookup Back Orders records based upon requested search criteria.

Parameters

records: (input) The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of Back Orders when WO ID's meets the search criteria, otherwise returns null.

Data Contract - WcfSI.UpdateOrderRequest

Name	Type / Size	Description
AllocateQnty	String	This displays the quantity allocated.
DeAllocateQnty	String	This displays the quantity deallocated.
ItemLineNum	String	This displays the item line number.
OrderID	String	This displays the order ID.

Strores Inventory- SIProduct

LookupProductMethod

WcfSI.SIOrderServvice.InsertandValidateOrderRecordmethod (WcfSI.InsertOrderRequest)
You use this method to lookup Stores Inventory Product records based upon requested search criteria.
Parameters

records: (input) Request. The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of SiProdMstr records when Product ID's meets the search criteria, otherwise returns null.

ValidateProductMethod

WcfSI.SIOrderServvice.ValidateProductMethod (WcfSI.InsertOrderRequest)
You use this method to validate a Stores Inventory Product ID in the SiProdMstr table. **Parameters**

records: (input) Request. The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns subset of SiProdMstr record when the Product ID is found, otherwise returns null.

Strores Inventory- SITransaction

LookupTransactionMethod

WcfSI.SITransactionSerrvice.LookupTransactionMethod (WcfSI.LookupTransactionRequest)
You use this method to lookup Stores Inventory Transaction records based upon requested search criteria.
Parameters

records: (input) Request. The request includes the maximum number of rows to return and an array of parameters where each additional parameter narrows the focus of the lookup. At least one parameter must be supplied.

returns: Returns an array of SiProdMstr records when Product ID's meets the search criteria, otherwise returns null.

TimeClock Plus Interface

TCPGetTimecards

This service returns timecard records that have created with this API. This can be run for all employees or for a single employee. In addition to the information inserted when the record was created, the date and time of the insertion will be returned.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri =
"http://localhost/FusionOSServices/v0/ONESolution/TimeClockPlus/Timecards";

      // append an employee id if desired.
    uri += "/E00001"

    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        while (uri != null)
        {
            string result = wc.DownloadString(uri);

            var response = JObject.Parse(result);

            var timecards =
(JContainer)response["RequestResponse"]["DATA"]["INTERFACETIMECARDS"];

            foreach (var timecard in timecards )
            {
                // Your code goes here
                string empId = (string)timecard["EMPID"];
                string hours = (string)timecard["HRS"];

            }
            // responses with more then 1000 positions are broken up into
multiple requests
            uri = (string)response["RequestResponse"]["NextUri"];
        }
    }
}
```

Sample Response: JSON

```
{
              "RequestResponse": {
                "Count": "2",
                "DATA": {
                        "INTERFACETIMECARDS":  [
                     { "EMPID": "E0001",
                       "HRS": "6.00000",
                       "CDH": "3001",
                       "TRANDT": "5/23/2017 4:24:58 PM",
                         "CREATEWHEN": "5/25/2017 8:36:02 AM"
                     },
                     { "EMPID": "E0002",
                       "HRS": "1.00000",
                       "CDH": "3050",
                       "TRANDT": "5/23/2017 4:24:58 PM"
                         "CREATEWHEN": "5/25/2017 8:36:02 AM"
                     }

                  }
              }
            }
```

TCPPutTimecards

This service writes time card info to Finance Enterprise.

Sample Code: C#

```
using System.Net;
using Newtonsoft.Json.Linq;

public void MethodName(parms){

    string uri =
"http://localhost/FusionOSServices/v0/ONESolution/TimeClockPlus/Timecards";
    string data = "[" +
                  "  {" +
                  " \"EmpId\": \"E00011\", " +
                  " \"TranDt\" : \"3/29/2017\", " +
                  " \"RecType\" : \"S0\", " +
                  " \"Hrs\": 4," +
                  " \"Cdh\" : \"3001\", " +
                  "  } " +
                  " , " +
                  "  { " +
                  "  \"EmpId\": \"E00011\", " +
                  "  \"TranDt\" : \"3/15/2017\", " +
                  "  \"RecType\" : \"PM\", " +
                  "  \"Hrs\": 4.00000,  " +
                  "  \"Cdh\" : \"3001\", " +
                  "  } " +
                  "  ]";

    using (WebClient wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        // Replace "ID" with supplied AppID
        wc.Headers.Set("X-APPID", "ID");
        // Replace "KEY" with supplied AppKey
        wc.Headers.Set("X-APPKEY", "KEY");

        string result = wc.UploadString(uri, "PUT", data);

        var response = JObject.Parse(result);
        var count = (string)response["RequestResponse"]["Count"];

        // verify count is expected number
        }
    }

}
```

Sample Response: JSON

```
{
    "RequestResponse": {
        "Count": "2",
        "DATA": null
    }
}
```

Workflow – Mobile Service

FetchMethod

WcfWF.WorkflowMobileService.FetchMethod
You use this method to fetch the data for the current user.

Parameters

records:
returns:

Workflow Service

You use this service to access the workflow.

GetHistoryRecordsMethod

WcfWF.WorkflowServices.GetHistoryRecordsMethod
You use this method to get all the history records for the specific instance.

Parameters

records:	(input) Key, Model, and Version. The instance key, model ID, and version number.
returns:	Returns the list of history records.

GetInstanceDetailMethod

WcfWF.WorkflowServices.GetInstanceDetailMethod
You use this method to get the workflow instance details.

Parameters

records:	(input) UniqueInstance Key, ModelID, and Version. The unique instance key, model ID, and version number.
returns:	

GetModelsForRoleTaskMethod

WcfWF.WorkflowServices.GetModelsForRoleTaskMethod
You use this method to get the workflow models for the user and their roles.

Parameters

records:
returns:

GetPendingAttachmentsMethod

WcfWF.WorkflowServices.GetPendingAttachmentsMethod
You use this method to get the document summary information about the documents that are pending attachments for the given mask.

Parameters

records:	(input) The mask of the screen we're getting the documents for.
returns:	An array of the summary information about the documents that are pending attachments.

GetTaskListDataMethod

WcfWF.WorkflowServices.GetTaskListDataMethod
You use this method to get the task list data for the specific record.

Parameters

records:	(input) The instance key to fetch the task list for.
returns:	The task list data for the specific record.

GetUserOutOfOfficeMethod

WcfWF.WorkflowServices.GetUserOutOfOfficeMethod
You use this method to get user out of office status. If the current user needs to use this method, set the user id to be blank.

Parameters

records:	(input) UserID
returns:	

GetWorkflowModelsMethod

WcfWF.WorkflowServices.GetWorkflowModelsMethod
You use this method to get the workflow models for the user.

Parameters

records:
returns:

GetWorkflowTasksMethod

WcfWF.WorkflowServices.GetWorkflowTasksMethod
You use this method to get the workflow tasks for a given model.

Parameters

records:	(input) ModelID, Version, Include Roles, and ShowAdminView.
returns:	

GetWorkflowUsersMethod

WcfWF.WorkflowServices.GetWorkFlowUsersMethod
You use this method to get the user list to select for delegation of Workflow task.

Parameters
records:
returns:

ProcessTaskActionMethod

WcfWF.WorkflowServices.ProcessTaskActionMethod
You use this method to process workflow tasks for the model.

Parameters

records:	Workflow Action
returns:	

SetUserOutOfOfficeMethod

WcfWF.WorkflowServices.SetUserOutOfOfficeMethod
You use this method to set user out of office. If the current user needs to used, set the user id to be blank.

Parameters
records:
returns:

Work Order

DeleteWorkOrderRecords

WcfWO.WorkOrderService.DeleteWorkOrderRecords (WcfWO.WorkOrderRequest)
You use this method to delete one or more work order records.

Parameters

records:	(input) RequestArray
returns:	True if no errors, otherwise false.

Data Contract - WcfWo.WorkOrderRequest

Name	**Type / Size**	**Description**
EndDt	DateTime	This displays the end date.
GlGr	String	
LongDesc	String	This displays the long description.
ShortDesc	String	This displays the long description.
StartDt	DateTime	This display the start date.
WoNo	String	This displays the work order number.

InsertWorkOrderRecrods

WcfWO.WorkOrderService.InsertWorkOrderRecords (WcfWO.WorkOrderRequest)
You use this method to inserts one or more work order records.

Parameters

records:	(input) RequestArray
returns:	True if no errors, otherwise false.

Data Contract - WcfWo.WorkOrderRequest

Name	Type / Size	Description
EndDt	DateTime	This displays the end date.
GlGr	String	
LongDesc	String	This displays the long description.
ShortDesc	String	This displays the long description.
StartDt	DateTime	This display the start date.
WoNo	String	This displays the work order number.

UpdateWorkOrderRecrods

WcfWO.WorkOrderService.UpdateWorkOrderRecords (WcfWO.WorkOrderRequest)
You use this method to inserts one or more work order records.

Parameters

records: (input) RequestArray

returns: True if no errors, otherwise false.

Data Contract - WcfWo.WorkOrderRequest

Name	Type / Size	Description
EndDt	DateTime	This displays the end date.
GlGr	String	
LongDesc	String	This displays the long description.
ShortDesc	String	This displays the long description.
StartDt	DateTime	This display the start date.
WoNo	String	This displays the work order number.