INEIGHT

Master Data – Employees

Integration Specification

InEight®
PLATFORM

Changelog

This changelog contains only significant or other notable changes to the document revision. Editorial or minor changes that do not affect the context of the document are not included in the changelog.

Rev	Date	Description
2.0	06-JUNE-2019	Copied from original template Added draft data for Employees_Patch Published revision.
3.0	02-OCT-2019	Updated information for Employee_Patch Added Employees_Get Modified multiple descriptions of Employee fields for clarity. Updated Employees (Post), Employees (Patch), Employees (Get). Deprecated EmailAddress, WorkPhoneNumber, WorkCellPhoneNumber, DateFormat, DecimalFormat fields. Published revision.
4.0	15-SEP-2020	Updated UoMName in Crafts and Employees (Post) field descriptions. Field accepts either the Name or SourceSystemId of a valid UOM record. Added note in Union Payroll Indicators Sample JSON. Deprecated Language field from Employees (Post), Employees (Patch), Employees (Get). Published revision (Release 20.5)
5.0	29-JAN-2021	Added integrations for Contacts (Import) and Contacts (Get). Added new integration for Operational Resources. Published revision (Release 20.11).
6.0	03-APR-2021	Added integrations for Markets (Import) and Markets (Get). Published revision (Release 21.3).
7.0	06-JAN-2022	Updated to include integrations for UOM, General Ledger, Incoterms, and Payment Terms. Published revision (Release 21.12).
8.0	11-MAY-2022	Added VendorAddresses array and associated fields to Vendors. Published revision (Release 22.2).
9.0	12-JUL-2022	Added Status field to Vendors. Published revision (Release 22.4).
10.0	29-SEP-2023	Reviewed and update document for clarification. Published revision (Release 23.8).
11.0	07-OCT-2024	Deleted IsActive field and added Status field in General Ledger, Incoterms, and Payment Terms. Updated APIM Name to match what is shown in APIM. Published revision (Release 24.9).
12.0	29-MAY-2025	Updated Error messages, Fields, and Field descriptions sections for Trades, Crafts, Vendors, Employees (POST), Contacts, General Ledger, Incoterms, and Payment Terms.

©2025 InEight Inc. All rights reserved.

Contents

Overview **1**
Integrations in This Document 1
Relationships and Dependencies 3
Employee or User 3
Employees, Trades, and Crafts 3
Vendors 5
Unions 5
Union Payroll Indicators 6
Accounting Ledger 6
Incoterms 6
Payment Terms 7
Prerequisites 7
Units of Measure and Currency 8
Payroll Indicators 9
Trades **10**
Fields 10
Field Descriptions 10
Error Messages 11
Sample JSON 12
Verification 12
Crafts **13**
Fields 13
Field Descriptions 14
Error Messages 15
Sample JSON 17
Verification 17
Vendors **18**
Fields 18
Field Descriptions 19
Error Messages 21

Sample JSON 23
Verification 24
Unions 25
Fields 25
Field Descriptions 25
Error Messages 26
Sample JSON 27
Verification 27
Union Payroll Indicators 28
Fields 28
Field Descriptions 28
Error Messages 29
Sample JSON 29
Employees (Post) 31
Fields 31
Field Descriptions 32
Error Messages 35
Sample JSON 37
Verification 37
Employees (Patch) 39
Fields 39
Field Descriptions 39
Sample JSON 39
Verification 40
Employees (Get) 41
Supported Filters 41
Fields 41
Field Descriptions 41
Sample JSON 42
Verification 43
Contacts 44

Contacts (Import) 44
Contacts (Get) 44
Fields 45
Field Descriptions 45
Error Messages 46
Sample JSON 47
Verification 48
Markets 49
Markets (Import) 49
Markets (Get) 49
Fields 50
Field Descriptions 50
Sample JSON 50
Verification 51
Currencies 52
Currencies (Get) 52
Fields 52
Field Descriptions 52
Sample JSON 53
Verification 53
Units of Measure 55
UOMs (Get) 55
Fields 55
Descriptions 55
Sample JSON 56
Verification 57
General Ledger 58
Fields 58
Field Descriptions 58
Error Messages 59
Sample JSON 60

Verification 60
Incoterms 61
Fields 61
Field Descriptions 61
Error Messages 62
Sample JSON 63
Verification 63
Payment Terms 64
Fields 64
Field Descriptions 64
Error Messages 65
Sample JSON 66
Verification 66

Overview

Integrations in This Document

Integration	Description
Trades	The Trades Integration allows InEight customers to provide a complete list of trades or individual records as maintenance of the master list. Trades are the top tier of the Employee hierarchy (Trade > Craft > Employee) and are used to filter for either a craft or specific employee resources.
Crafts	The Crafts Integration allows InEight customers to provide a complete list of crafts or individual records as maintenance of the master list. Crafts are the middle tier of the Employee hierarchy (Trade > Craft > Employee) and are used to assign general categories of resources that may be needed on a Work Plan or to filter on specific employee resources.
Vendors	The Vendors (Import) integration allows customers to maintain a master list of all the Vendors that can be used in any Project. The Vendors (Get) integration provides the full list of Vendors available for use within the customer environment.
Unions	This integration allows a customer to maintain a master list of Unions in InEight. Unions can be associated with Employees and Crafts at either the Account (all projects), or specified Project levels. These associations can be created and maintained either in their own (separate) integrations, or through the InEight UI.
Union Payroll Indicators	This integration allows an external system to maintain an association of specific Payroll Indicators created in InEight Plan to a Union and/or a specific Employee. This determines what Payroll Indicators are valid by a given Union or specific Employee, utilized in Time Center.
Employees (Post)	This integration allows InEight customers to provide a list of all employee resources for their entire organization, and to maintain the list as employees are hired, terminated, or details of their employment change.
Employees (Patch)	Provides a method to update Employee records without providing details for all fields. Only the unique record identifier and the specific fields that should be updated will need to be provided. This integration is useful when there is specific content of Employee records in InEight that are not managed by a customer's HR or ERP system. For example, if Employee PINs are solely managed in InEight and updating an entire Employee record using the Employees (Post) integration will cause an existing PIN to be nulled, the Employee Patch integration can be used to update only specific parts of Employee records and leave the InEight managed PIN intact.
Employees (Get)	This integration provides all Employee records created in InEight for use in any project.
Contacts	These integrations allow a customer to import and export information for their master list of contacts from their internal systems into the InEight cloud platform and from the InEight system to external systems. The integrations allow customers to create and maintain their master list of contacts, which are used for various business functions in multiple InEight cloud applications.

Integration	Description
Markets	These integrations allow a customer to import and export information for their master list of markets from their internal systems into the InEight cloud platform and from the InEight system to external systems. The integrations allow customers to create and maintain their master list of markets, which are used for various business functions in multiple InEight cloud applications.
Currencies	Provides the full list of currencies available for use within the customer environment.
Unit of Measure	Provides the full list of the units of measure available for use within the customer environment.
General Ledger	This integration allows a customer to import general ledger accounts/records that are used to track and record financial transactions.
Incoterms	This integration allows a customer to import Incoterms® values for sale of goods in international transactions.
Payment Terms	This integration allows a customer to import payment terms, which determine when a vendor must be paid, period between payments, discounts when applicable, any exceptions, etc.

Relationships and Dependencies

Employee or User

Within the system, anyone can be a user, an employee, or both a user and an employee. Employees are resources in the InEight system that are listed as operational resources and can be assigned to work packages, daily plans, or weekly timesheets. For example, when creating a daily plan in the Plan application an employee can be assigned to work items and time can be recorded for them. Users are different from employees in that they have specific login credentials and are licensed to utilize functionality within InEight applications. Users who do not also have an associated Employee record are not resources in InEight and cannot be assigned to work items.

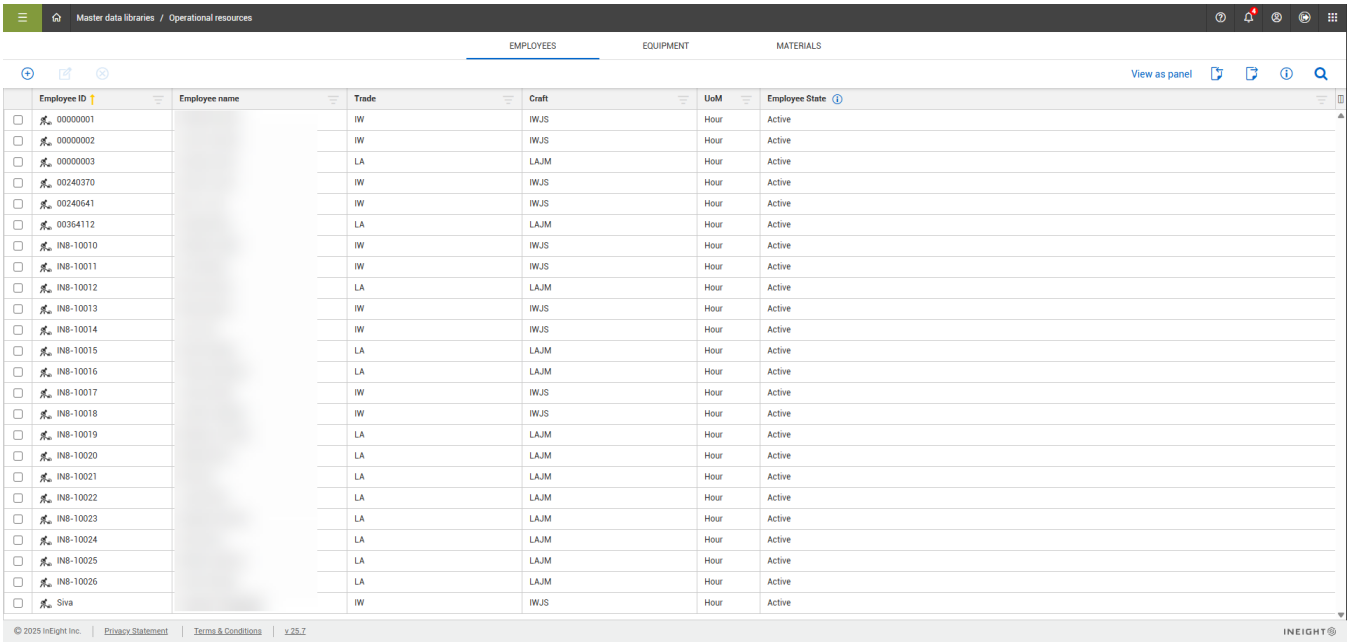
Employees, Trades, and Crafts

In the InEight system, employees can be categorized by the type of work they perform (trade) and their proficiency level for that work (craft). This categorization is hierarchical and only allows 1:1 assignments of trade to craft, and craft to employee. The table shows examples of the relationships between trades and crafts.

Trade	Craft
Carpenter	• Journeyman Carpenter • Carpenter I • Carpenter II • Apprentice Carpenter
Pipefitter	• Journey Pipefitter • Pipefitter II
Equipment Operator	• Class II Operator • Crane Operator • Bulldozer Operator
Steelworker	• Apprentice Steelworker • Steelworker III • Master Steelworker

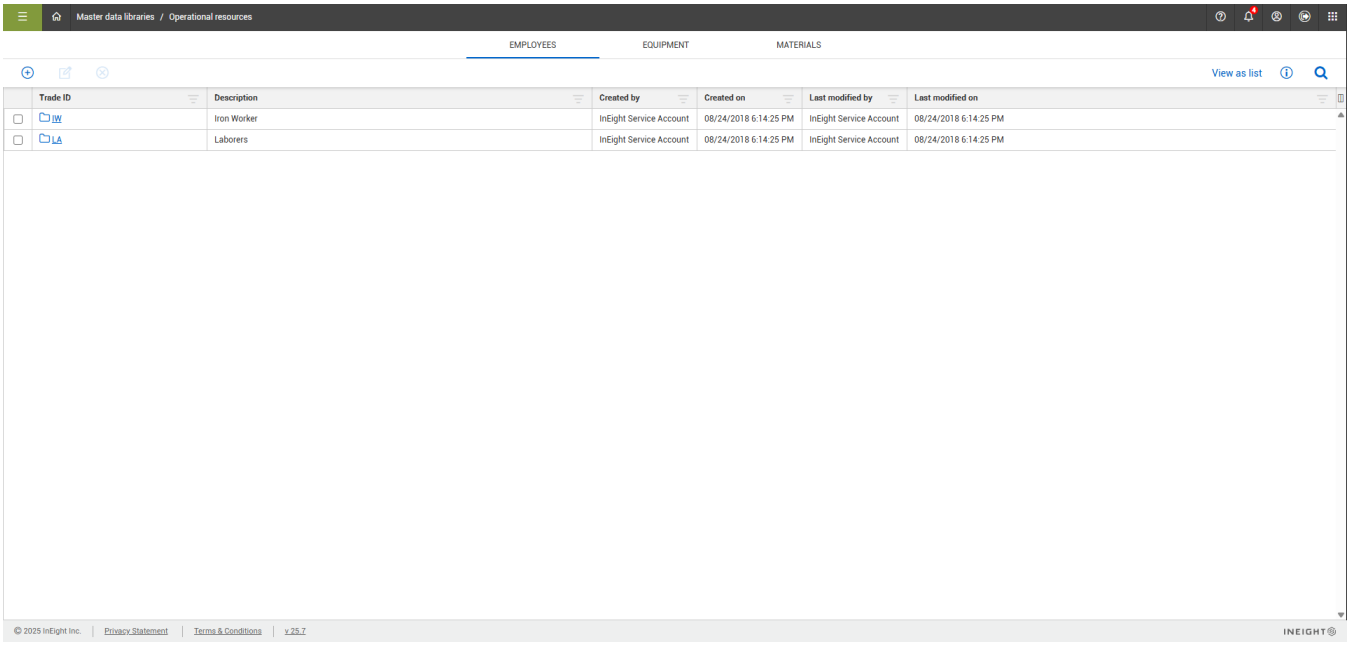
To support the hierarchy, trades must be present before crafts, and crafts must be present before employees. Unions and vendors are optional associations to employees, but also must be present prior to employees to create the association through integration.

Import trades into the cloud platform using the API Upsert Trades, or manually create them through the Operational Resources UI.



Operational resources (Trade) view in resource library

Import all the crafts and associate them to their respective trades in the cloud platform using the API Upsert Crafts, or manually create crafts in the Operational Resources UI.



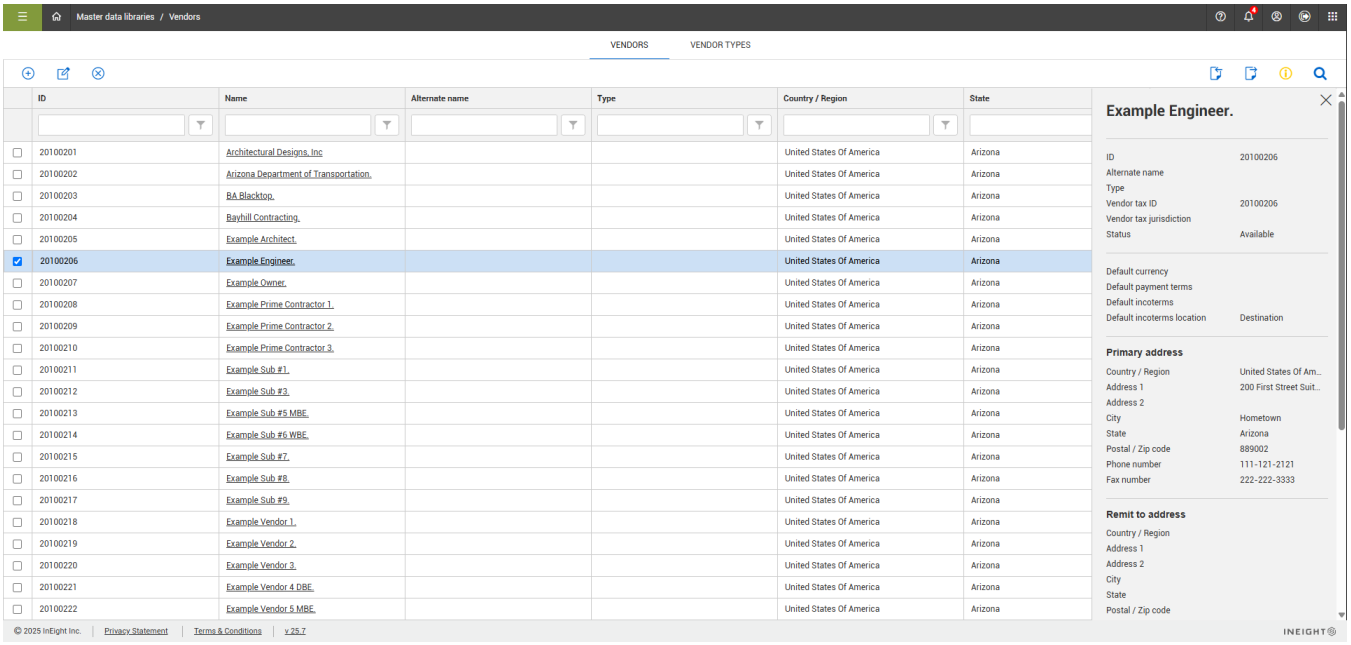
Operational resources (Craft) view in resource library

Vendors

Individual employee records can optionally be associated to a Vendor. When this association exists, a user (also associated to a vendor) creating a work plan can only select employees for assignment that match their vendor code.

Vendors can also be assigned to a craft. Like employees, when this association exists, users creating work plans can only use crafts that match the user's vendor code.

Import the vendor list using API Upsert Vendors, or manually create vendors in the Resource Library UI.



Vendors view in resource library

Unions

Individual Employee records can be optionally associated to a union. This association drives the default assignment of a union for the employee within projects and estimated hourly pay based on Craft records assigned to a union.

When an employee is assigned to a union, their craft assignment must also be assigned to the same union, otherwise the craft cannot be assigned to the employee. Unions can also be assigned to a craft. This association limits the use of that craft to the union it was assigned.

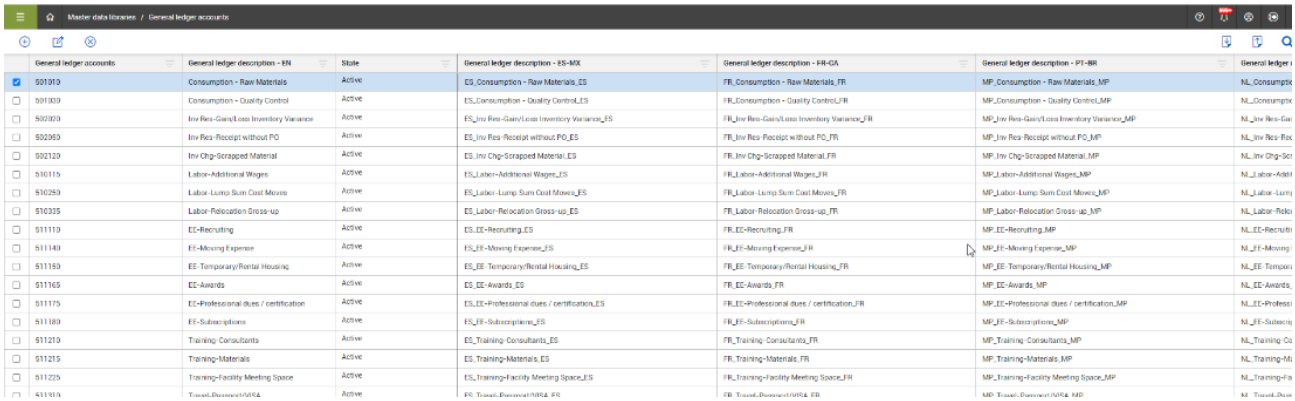
Union Payroll Indicators

Payroll indicators provide optional encoding that can be used by a payroll system to identify attendance, premiums, and extra pay. Specific payroll indicators that have been pre-defined can be directly associated to a union and/or an employee. In both cases, the association limits which indicators can be used on a Daily Plan at the intersection of an employee and a time record for the day. When associated to a union, the limitation is based on the union to which the employee is assigned. When associated directly to the employee, the limitation is based on the direct assignment although, you can assign payroll indicators to the employee at the project level.

Currently, there is nowhere in the UI to make the assignment of payroll indicators to unions, so it must be done via the Upsert UnionPayrollIndicators API. Without making these assignments, employees will have an error in Time Center whenever timecard records are submitted with extra pay or premiums assigned to them on a daily plan that are not listed on their employee or union record.

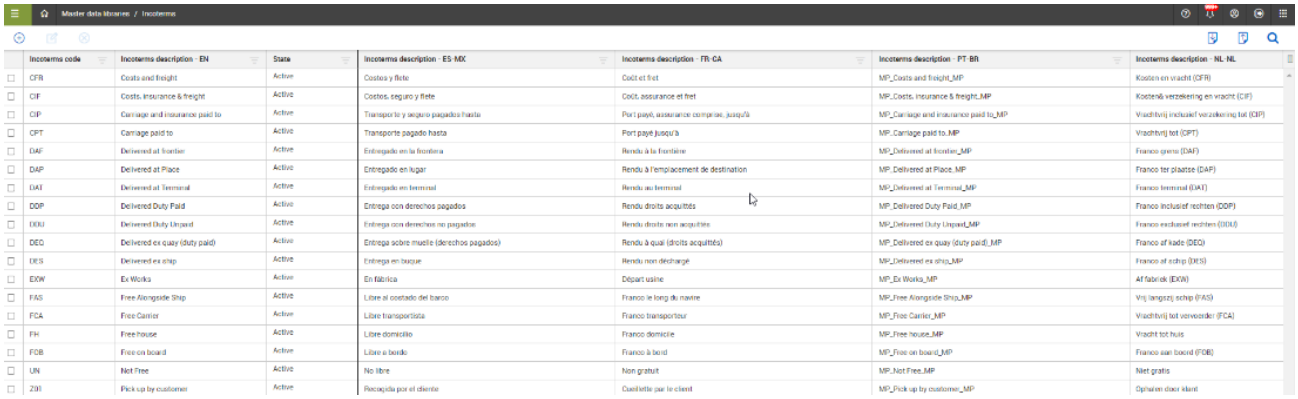
Accounting Ledger

A general ledger (GL) is a set of numbered accounts a business uses to keep track of its financial transactions and to prepare financial reports. A chart of accounts lists all the accounts in the general ledger, which can number in the thousands for a large business. Using this API, users can import their GL accounting records into the InEight cloud platform. The general ledger accounts are used on InEight Contract and InEight Control line items.



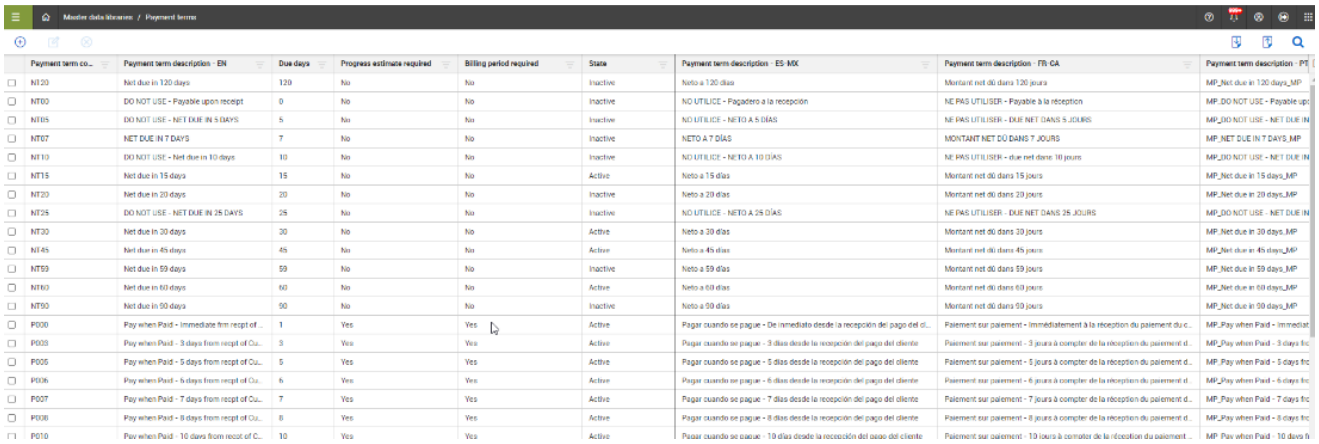
Incoterms

The Incoterms® are a set of individual rules issued by the International Chamber of Commerce (ICC), which define the responsibilities of sellers and buyers for the sale of goods in international transactions. Of primary importance is that each Incoterms rule clarifies the tasks, costs, and risks to be borne by buyers and sellers in these transactions. Familiarizing yourself with Incoterms will help improve smoother transaction by clearly defining who is responsible for what and each step of the transaction. Using this API, users can import their IncoTerms rule records into the InEight cloud platform. The incoterms are used in InEight Contract to define the incoterms of each PO created through InEight Contract.



Payment Terms

Payment terms, explained as the terms which dictate when a vendor must be paid, vary in policy. Other businesses offer payment terms as a perk of becoming a client. These terms could be a pay in 30 days term, a 2% discount for paying within 10 days (2/1 net 30), and other terms which allow the customer to pay later. Using this API, users can import their Payment Terms records into the InEight cloud platform. The payment terms are used in InEight Contract to define the payment terms of each PO created through InEight Contract.



Prerequisites

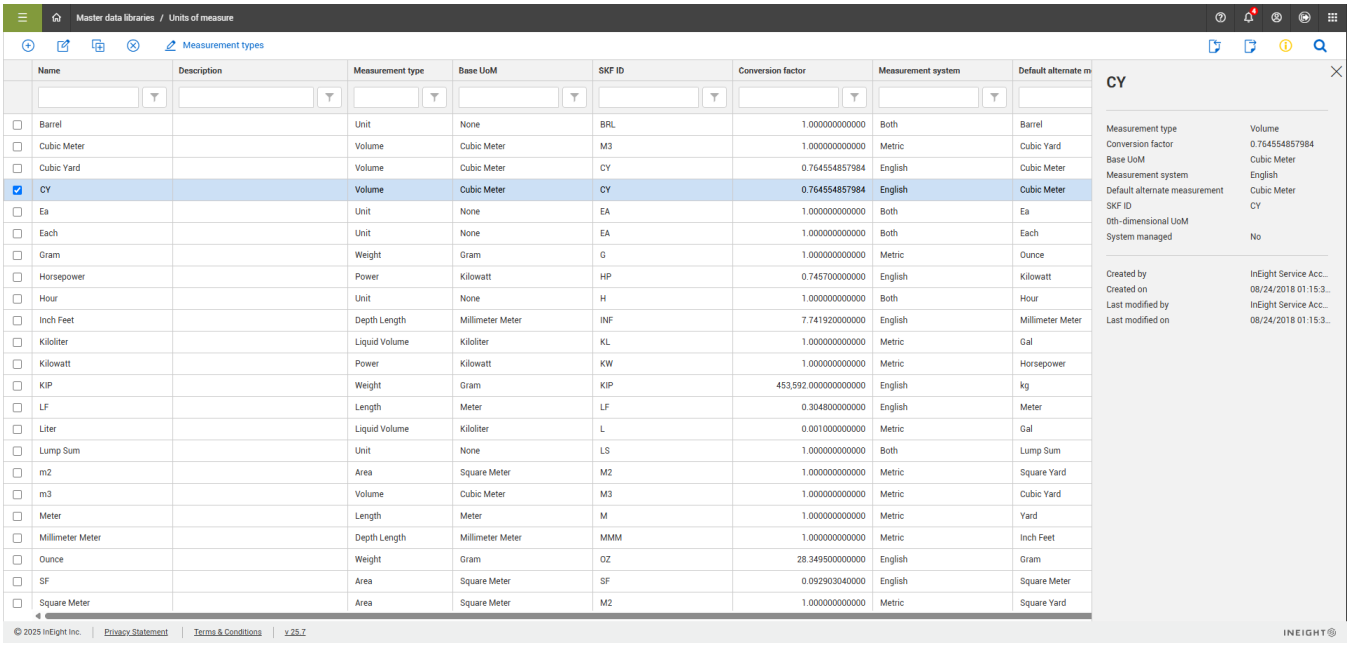
The following table lists prerequisites within InEight for using the integrations in this document. These can be the presence of required supporting data or system configurations. All items in the table represent required fields within the integrations. There are many other fields within the integrations that reference data from various areas of InEight but are optional and will not prevent the addition or update of a record. These optional fields are called out within the individual integration field descriptions.

Prerequisite	Description	Required by
Trade	Master Data referenced within integrated records	Crafts
Crafts	Master Data referenced within integrated records	Employees

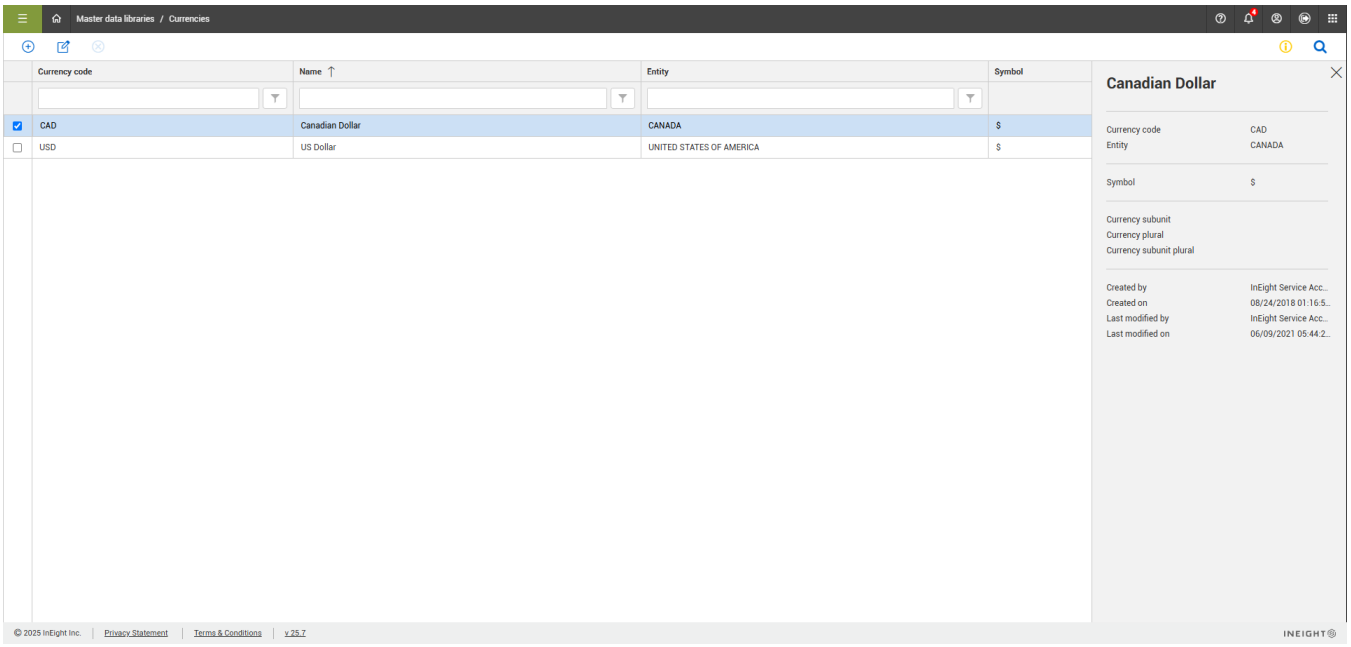
Prerequisite	Description	Required by
Payroll Indicators	Master Data referenced within integrated records. Must be manually created in organization Settings > Progress > Daily Planning > Others.	Union Payroll Indicators
Unions	Master Data referenced within integrated records	Union Payroll Indicators, Crafts, Employees
Currencies	Master Data referenced within integrated records	Employees

Units of Measure and Currency

The Unit of Measure (UoM) and Currency master data should be defined and available in InEight. The presence of data for these entities can be verified in the Account level master data section, which can be opened by clicking on the gear box as shown below.



Unit of measure master data view



Currency master data view

Payroll Indicators

Payroll indicators are defined in Master data libraries > **Payroll indicators** and associated with employees in Project Settings > Assigned payroll indicators.

Trades

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Possibly hundreds of records to allow for all types of work supported by a customer.
APIM Name		Upsert Trades
InEight Application	**Starting Version**	18.2
	Ending Version	

Fields

Depth	Name	Type	Precision[1]	Parent	Req.
1	TradeDisplay[2]	String	50		Yes
1	TradeDescription	String	255		No
1	IsActive	Boolean	NA		No
1	SourceSystemId	String	50		No
1	SourceSystemName	String	50		No

2 - Natural Key field.

Field Descriptions

Name	Description	Example
TradeDisplay	A code (usually 2 or 6 characters) that represents or describes the Trade of an employee. TradeDisplay shows as the Trade Id field in the UI. This value must be a unique to a Trade.	AD
TradeDescription	Customer-defined field that is typically used to describe or provide details about the Trade.	Craft Administration
IsActive	Determines if the record is active and available to the system. If a value is not provided, the value *true* is used. Sending a value of *false* causes a soft delete of the record in InEight.	True
SourceSystemId	The identification of the employee trade used within the external system. In most cases, this matches the TradeDisplay field. This field is used by InEight to match employee records to other entities (e.g., Project Employee) where an employee is cross-referenced.	AD

Name	Description	Example
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	CLNTERPSOU

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • TradeDisplay • TradeDescription	400	The request is invalid. The [Field Name] Field is required.
API Validation	Payload exceeds String Length Size. [Field Name] options are: • TradeDisplay • TradeDescription • SourceSystemId • SourceSystemName	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length]. Message: The request is invalid.
API Validation	Payload data is Empty	400	Payload was missing or invalid and was not parsed.
Entity Logic Validation	Payload has more than one record with same SourceSystemId	200	Duplicate SourceSystemId {Comma separated list of SourceSystemIds} found. Aborting the process.
Entity Logic Validation	User does not have Add, Edit and Delete Permission	200	Ignoring the processing of the Trade with the Id {TradeId} due to absence of permissions: Add, Edit & Delete
Entity Logic Validation	Payload has more than one record with same DisplayId	200	Duplicate DisplayId {Comma separated list of DisplayIds} found. Aborting the process.
Entity Logic Validation	Payload has conflict with uniqueness on SourceSystemId or DisplayId with existing data	200	Ignoring Trade having display '{TradeDisplay}' as it is having duplicate SourceSystemId or DisplayId
Entity Logic Validation	Payload has inactive records while inserting	200	Ignoring inactive Trade '{TradeDisplay}' for adding

Entity Logic Validation	Deleting a Trade when it is associated with at-least one Craft.	200	Unable to delete Trade: '{TradeDisplay}', You must remove all associated crafts from this trade before it can be deleted

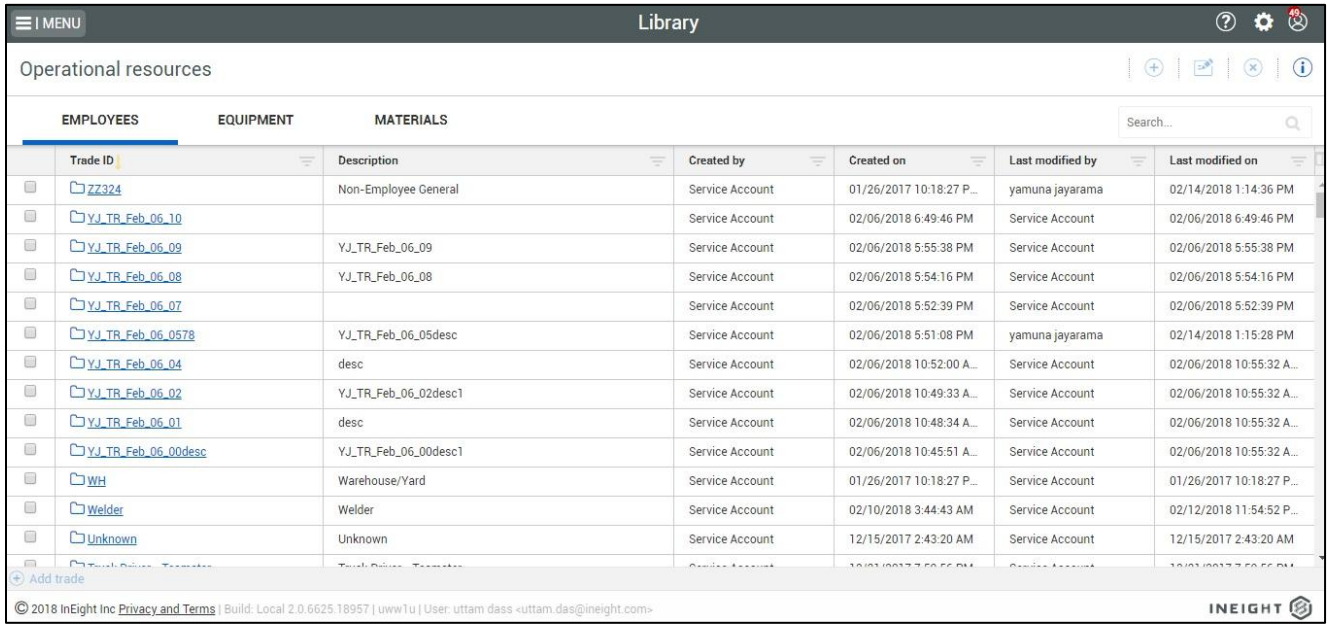
Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

Sample JSON

```
{
 "TradeDisplay": "AD",
 "TradeDescription": "Craft Administration",
 "IsActive": true,
 "SourceSystemId": "AD",
 "SourceSystemName": "CLNTERPSOU"
}
```

Verification

Import trades into InEight using the API Upsert Trades. The results can be validated from the Operational resources section in the InEight cloud platform. Refer below screenshot.



Operational resources (Trade) view in resource library

Crafts

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Possibly hundreds of records to allow for all types of work supported by a customer.
APIM Name		Upsert Crafts
InEight Application	**Starting Version**	18.2
	Ending Version	

Fields

Depth	Name	Type	Precision[1]	Parent	Req.
1	CraftId[2]	String	50		Yes
1	CraftDescription	String	255		No
1	TradeId	String	50		Yes
1	CurrencyCode	String	50		Yes
1	VendorCode	String	250		No
1	SourceSystemId	String	50		No
1	SourceSystemName	String	50		No
1	UnionSourceSystemId	String	50		No
1	UoMName	String	50		Yes
1	UseBaseWageFactors	Boolean	NA		No
1	StraightTimeRate	Number	16,5		No
1	OverTimeFactor	Number	16,5		No
1	OverTimeRate	Number	16,5		No
1	DoubleTimeFactor	Number	16,5		No
1	DoubleTimeRate	Number	16,5		No
1	IsActive	Boolean	NA		No

1 - For numeric data types, precision is given as total digits allowed in the field and the number of those digits that exist to the right of the decimal. For example, 16,5 represents a total of 16-digits allowed in the field with 5 of those digits existing as decimal places and 11 digits on the left of the decimal. The decimal is not counted as a digit.
2 - Natural Key field.

Field Descriptions

Name	Description	Example
CraftId	Unique, informal name of the Craft that is shown in the UI.	SA1S2
CraftDescription	Customer defined field that can be used to indicate the skill level and proficiency of the employee with associated craft.	Steamfitter Apprentice 1 Step 2 47%
TradeId	Code that represents or describes the trade of an employee associated with the Craft, typically 2 to 6 characters. Must match a valid Trade SourceSystemId.	AD
CurrencyCode	Identifies the currency type associated with the craft used to determine the standard rate of the craft. The value provided here must match a valid CurrencyCode in InEight master data.	USD
VendorCode	Identifies a vendor that exists in InEight. Crafts with a vendor code are managed by the associated vendor and availability is limited to that vendor. Must match a valid Vendor SourceSystemId.	000000001028
SourceSystemId	Identification of the employee craft used within the external system. In most cases, this matches the CraftId field. This field is used to match employee records to other entities (*e.g,.* Project Employee) where an employee is cross-referenced.	47PP
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	CLNTERPSOU
UnionSourceSystemId	An optional association of the craft to a specific union. The value sent in this field must match a valid Union in master data.	ILW
UoMName	Name or SourceSystemId of an existing Unit of Measure record that should be used for resources assigned to the craft.	HR
UseBaseWageFactors	Flag that determines whether to use the absolute values or the factors for pay rates.	False
StraightTimeRate	Rate for standard pay that should be applied for this craft within the assigned project.	35.00
OverTimeFactor	Factor used to multiply the standard rate of the employee to determine their overtime rate.	1.5
OverTimeRate	Exact calculated rate paid to the employee for overtime work as per applicable laws. (Straight time X Overtime Factor = Overtime)	35.00
DoubleTimeFactor	Factor used to multiply the standard rate of the employee to determine their Double time rate.	2
DoubleTimeRate	Exact calculated rate paid to the employee for double time work as per applicable laws. (Straight time X Double Time Factor = Double time)	35.00

Name	Description	Example
IsActive	Determines if the record is active and available to the system. If a value is not provided, the value *true* is used. Sending a value of *false* will cause a soft delete of the record in InEight.	True

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • CraftId • TradeId • CurrencyCode • UoMName	400	The request is invalid. The [Field Name] Field is required.
API Validation	Payload exceeds String Length Size. [Field Name] options are: • CraftId • CraftDescription • TradeId • CurrencyCode • VendorCode • SourceSystemId • SourceSystemName • UnionSourceSystemId • UoMName	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length].
API Validation	Payload data is empty	400	Payload was missing or invalid and was not parsed.
Entity Logic Validation	Payload has more than one record with same SourceSystemId/ CraftId	200	Duplicate SourceSystemId/DisplayId {Comma separated list} found. Aborting the process.
Entity Logic Validation	Payload has conflict with uniqueness on SourceSystemId or CraftId with existing data	200	Ignoring Craft '{CraftId}' as it is having duplicate SourceSystemId or DisplayId
Entity Logic Validation	TradeId does not exist in the database	200	Ignoring Craft '{CraftId}' as it is having invalid TradeId {TradeSourceSystemId}
Entity Logic Validation	CurrencyCode does not exists in the database	200	Ignoring Craft '{CraftId}' as it is having invalid {CurrencyCode}

API/Entity Logic	Condition	Code	Message
Entity Logic Validation	UOMName does not exists in the database	200	Ignoring Craft '{CraftId}' as it is having invalid {UOMName}
Entity Logic Validation	VendorSourceSystemId does not exists in the database	200	Ignoring Craft '{CraftId}' as it is having invalid {VendorSourceSystemId}
Entity Logic Validation	UnionSourceSystemId does not exists in the database	200	Ignoring Craft '{CraftId}' as it is having invalid {UnionSourceSystemId}
Entity Logic Validation	Payload has inactive records while inserting.	200	Ignoring inactive Craft '{CraftId}' for adding
Entity Logic Validation	Deleting a craft with employees or if it is associated to at-least one project.	200	Ignoring Craft '{CraftId}' for deleting as it is associated with employees and projects. So, You must remove all associated projects and employees from this craft before it can be deleted.
Entity Logic Validation	Changing the craft's vendor when the craft is associated to at least one project.	200	Ignoring Craft '{CraftId}' for updating because Vendor cannot be changed when an employee in this craft is assigned to at least one project.
Entity Logic Validation	Changing the craft's vendor when craft has employees.	200	Ignoring Craft '{CraftId}' for updating because Vendor cannot be changed if craft has employees.
Entity Logic Validation	Changing craft's currency when it has employees assigned to at-least one project.	200	Ignoring Craft '{CraftId}' for updating because Currency cannot be changed when an employee in this craft is assigned to at least one project.
Entity Logic Validation	StraightTimeRate is null and OverTimeFactor / DoubleTimeFactor / OverTimeRate / DoubleTimeRate are not null	200	Ignoring Craft '{CraftId}' because ST rate is not passed.
Entity Logic Validation	Any of the TimeFactor or TimeRate is a negative value.	200	Ignoring Craft '{CraftId}' because Rates/Factors should not accept -ve value.

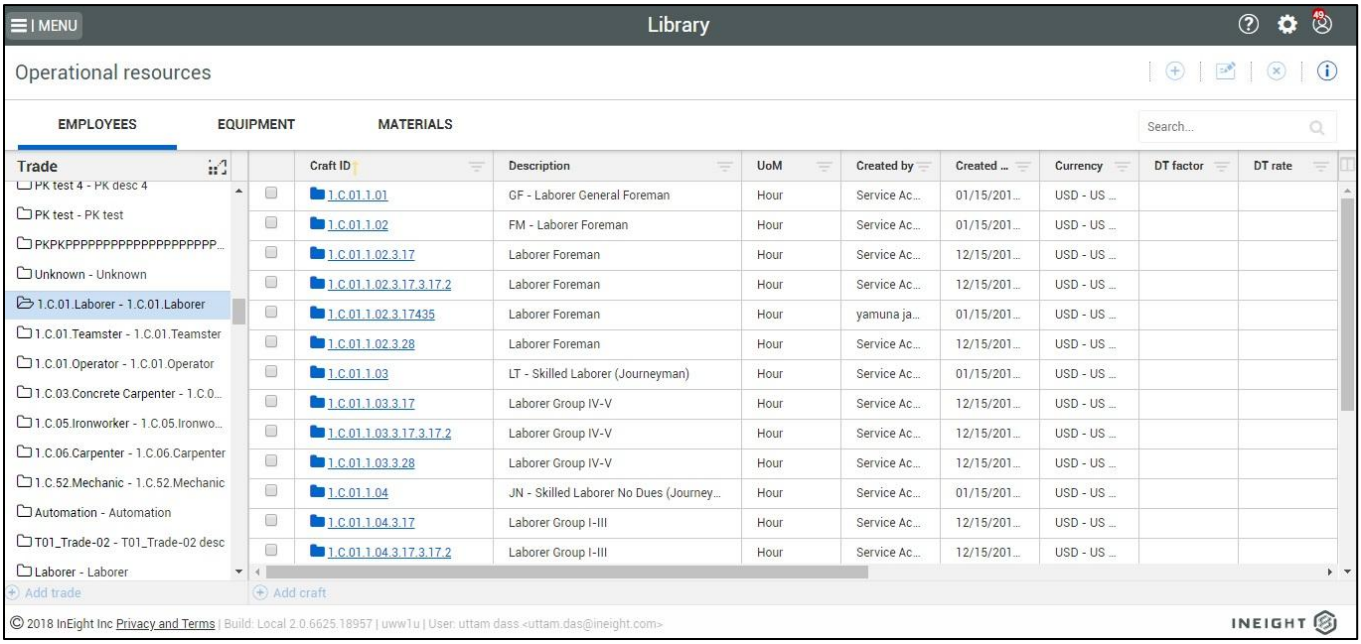
Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

Sample JSON


```
[
 {
 "CraftId": "Craft_Jan_22_07",
 "CraftDescription": "12345678",
 "TradeId": "00 SSID",
 "CurrencyCode": "USD",
 "VendorCode": "0000001028",
 "SourceSystemId": "Craft_Jan_22_07",
 "SourceSystemName": "CLNTERPSSN",
 "UnionSourceSystemId": "SSId47",
 "UoMName": "Acre",
 "UseBaseWageFactors": false,
 "StraightTimeRate": 35.00,
 "OverTimeFactor": 1,
 "OverTimeRate": 35.00,
 "DoubleTimeFactor": 1,
 "DoubleTimeRate": 35.00,
 "IsActive": true
 }
]
```


Verification

Import all the crafts and associate them to their respective trades in the InEight cloud platform using the Upsert Crafts API.



Operational resources (Craft) view in resource library

Vendors

Direction		To InEight cloud platform from external system.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		Upsert Vendors
InEight Application	**Starting Version**	18.2
	Ending Version	

Direction		From InEight cloud platform to external system.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		List Vendors
InEight Application	**Starting Version**	19.9
	Ending Version	

Fields

Depth	Name	Type	Precision	Parent	Req.
1	SourceSystemId	String	250		No
1	SourceSystemName	String	50		No
1	VendorCode	String	250		Yes
1	VendorDescription	String	250		Yes
1	IsActive	Boolean	NA		No
1	AlternateName	String	250		No
1	Address1	String	250		No
1	Address2	String	250		No
1	ZipCode	String	50		No
1	City	String	100		No

Depth	Name	Type	Precision	Parent	Req.
1	RegionISOCode	String	10		No
1	CountryISOCode	String	10		No
1	TelephoneNumber	String	50		No
1	FaxNumber	String	50		No
1	PaymentTermsSourceSystemId	String	250		No
1	INCOTermsSourceSystemId	String	250		No
1	DefaultINCOTermsLocation	String	50		No
1	VendorTypeReference	String	50		No
1	DefaultCurrencyCode	String	50		No
1	PreferredLanguage	String	5		No
1	VendorTaxJurisdiction	String	60		No
1	Status	String	50		No
1	VendorTaxId	String	60		No
1	VendorAddresses	String	Array		
2	VendorAddressType	String	50	VendorAddresses	
2	Address1	String	250	VendorAddresses	
2	Address2	String	250	VendorAddresses	
2	Zipcode	String	50	VendorAddresses	
2	City	String	100	VendorAddresses	
2	RegionISOCode	String	10	VendorAddresses	
2	CountryISOCode	String	10	VendorAddresses	
2	TelephoneNumber	String	50	VendorAddresses	
2	FaxNumber	String	50	VendorAddresses	

Field Descriptions

Name	Description	Example
SourceSystemId	Identifier created by the system of record and used in all system communications as the primary method of specifying a unique record. This field does not display in normal use of InEight applications and can contain internal identifiers such as a GUID that is not understandable by typical system users.	0010079210
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	D365

Name	Description	Example
VendorCode	Unique code that shows as the Vendor ID in the UI.	0010079210
VendorDescription	User friendly description of the record that will be shown in normal use of InEight applications to assist users when performing searches or selections. This field supports a large number of text characters, but it is recommended to keep the description as short as possible while retaining an easily recognized meaning.	My Test Vendor
IsActive	Determines if the record is active and available to the system. If a value is not provided, the value *true* is used. Sending a value of *false* causes a soft delete of the record in InEight.	true
AlternateName	An alternate name that can be assigned to the vendor to aid searches.	Mitchell, Schmidt, & Hopkins
Address1	Street address of the vendor.	123 S. South Street
Address2	An optional second line for the street address of the vendor.	Suite 123
ZipCode	Postal or zip code of the vendor.	90210
City	City where the vendor is located.	Los Angeles
RegionISOCode	ISO 3166-2 Code for the region of the vendor.	US-CA
CountryISOCode	ISO Alpha-2 code for the country of the vendor.	US
TelephoneNumber	Telephone number for the vendor.	555-555-1212
FaxNumber	Fax number for the vendor	555-555-1212
PaymentTermsSourceSystemId	Optional payment terms that will be defaulted when the vendor is added to a contract. If used, data provided in this field will be validated against the Master Data Payment terms values.	NET15
INCOTermsSourceSystemId	Optional International Commercial (INCO) Terms that will be defaulted when the vendor is added to a contract. If used, data provided in this field will be validated against Master Data Incoterms values.	FCA
DefaultINCOTermsLocation	Optional location that risk passes from seller to buyer per the specified INCO Terms	Place of Delivery
VendorTypeReference	Optional reference to a Vendor Type. If used, data provided in this field must match a valid record in the Vendor Type master data located in Platform.	Z001
DefaultCurrencyCode	Not used. Reserved for future functionality.	Not applicable
PreferredLanguage	Not used. Reserved for future functionality.	Not applicable
VendorTaxJurisdiction	An optional Tax Jurisdiction Code that can be included when sending Invoice and Pay Requests from Contracts to an external system. This field can also be optionally used to when requesting tax details from an external taxation system for contract line items. InEight does not validate the data sent in this field.	250221105

Name	Description	Example
Status	Status of the vendor, either available or unavailable.	Available
VendorTaxId	Optional Tax Identification number issued for the vendor from a government tax authority (typically IRS for US Vendors) The value for this field must be unique per Vendor – Each Vendor must have their own Tax ID.	00-101010101
VendorAddresses	Array header for additional vendor address locations.	
VendorAddressType	Indicates the type of address for the vendor, which can be PRIMARY, REMIT TO, or ADDITIONAL.	PRIMARY
Address1	Street address of the vendor.	123 Cedar Street
Address2	An optional second line for the street address of the vendor.	Suite 200
Zipcode	Postal or zip code of the vendor.	06405
City	City where the vendor is located.	Branford
RegionISOCode	ISO 3166-2 Code for the region of the vendor.	US-CT
CountryISOCode	ISO Alpha-2 code for the country of the vendor.	US
TelephoneNumber	Telephone number for the vendor.	999-99-9999
FaxNumber	Fax number for the vendor	999-99-9999

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • VendorCode • VendorDescription	400	The request is invalid. The [Field Name] Field is required.

API/Entity Logic	Condition	Code	Message
API Validation	Payload exceeds String Length Size. [Field Name] options are: • SourceSystemId • SourceSystemName • VendorCode • VendorDescription • AlternateName • Address1 • Address2 • ZipCode • City • RegionISOCode • CountryISOCode • TelephoneNumber • FaxNumber • DefaultPaymentTerms • DefaultINCOTerms • DefaultINCOTermsLocation • VendorTypeReference • VendorTaxJurisdiction • VendorTaxId	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length]. Message: The request is invalid.
API Validation	Payload data is Empty	400	Payload was missing or invalid and was not parsed.
Entity Logic Validation	Payload has more than one record with same SourceSystemId	200	Duplicate SourceSystemId {Comma separated list of SourceSystemIds} found. Aborting the process.
Entity Logic Validation	User does not have Add, Edit and Delete Permission	200	Ignoring the processing of the Vendor with the Id {VendorId} due to absence of permissions: Add, Edit & Delete
Entity Logic Validation	Payload has more than one record with same DisplayId	200	Duplicate DisplayId {Comma separated list of DisplayIds} found. Aborting the process.
Entity Logic Validation	Payload has inactive records while inserting	200	Ignoring inactive Vendor '{VendorCode}' for adding
Entity Logic Validation	Payload Data has invalid Language	200	Ignoring Vendor '{VendorCode}' as it is having invalid PreferredLanguage '{LanguageId}'
Entity Logic Validation	Payload Data has invalid Currency	200	Ignoring Vendor '{VendorCode}' as it is having invalid DefaultCurrencyCode '{CurrencyCode}'
Entity Logic Validation	Payload Data has invalid VendorType	200	Ignoring Vendor '{VendorCode}' as it is having invalid VendorTypeReference '{VendorTypeName}'
Entity Logic Validation	Payload Data has invalid CountryISOCode	200	Ignoring Vendor '{VendorCode}' as it is having invalid CountryISOCode '{CountryISOCode}'
Entity Logic Validation	Payload Data has invalid RegionISOCode	200	Ignoring Vendor '{VendorCode}' as it is having invalid RegionISOCode '{RegionISOCode}'
Entity Logic Validation	Payload Data has VendorTaxId that is already used for another Vendor	200	Ignoring Vendor '{VendorCode}' as it is having duplicate VendorTaxId '{VendorTaxId}'

API/Entity Logic	Condition	Code	Message
Entity Logic Validation	Payload Data has invalid VendorAddressType	200	Ignoring Vendor '{VendorCode}' as it is having Invalid VendorAddressType '{VendorAddressType}'
Entity Logic Validation	Payload Data has invalid Vendor Status	200	Ignoring Vendor '{VendorCode}' as it is having invalid Status '{VendorStatusName}'
Entity Logic Validation	Payload Data has invalid PaymentTerm	200	Ignoring Vendor '{VendorCode}' as it is having invalid PaymentTerm '{PaymentTermSourceSystemId}'
Entity Logic Validation	Payload Data has invalid PaymentTerm	200	Ignoring Vendor '{VendorCode}' as it is having invalid PaymentTerm '{PaymentTermSourceSystemId}'
Entity Logic Validation	Payload Data has invalid IncoTerm	200	Ignoring Vendor '{VendorCode}' as it is having invalid IncoTerm '{IncoTermSourceSystemId}'
Entity Logic Validation	Deleting vendor that is associated in at-least one entity	200	Ignoring vendor '{VendorCode}' for deleting because it may be associated with projects, users, contacts, crafts, model-types, pieces of equipment and employees. So, you must remove all vendor associations before it can be deleted.

Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

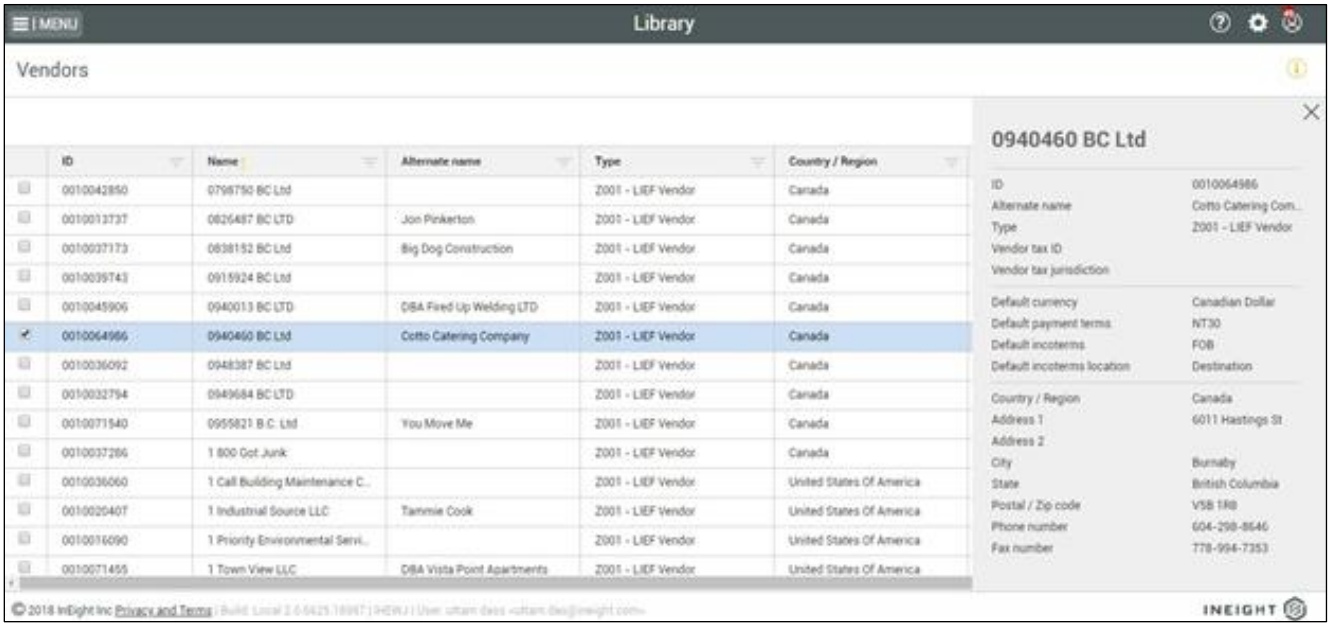
Sample JSON


```
{
"SourceSystemId": "23977438",
"SourceSystemName": "D365",
"VendorCode": "23977438",
"VendorDescription": "Fake Subcontractor, LLC",
"IsActive": true,
"AlternateName": "FSL",
"Address1": "123 S. A Street",
"Address2": "Suite 123",
"ZipCode": "85250",
"City": "Scottsdale",
"RegionISOCode": "US-AZ",
"CountryISOCode": "US",
"TelephoneNumber": "555-555-5555",
"FaxNumber": "555-555-5556",
"DefaultPaymentTermsSourceSystemId": "NET30",
"DefaultINCOTermsSourceSystemId": "FOB",
"DefaultINCOTermsLocation": "Port of Shipment",
"VendorTypeReference": "Z001",
"DefaultCurrencyCode": "USD",
"PreferredLanguage": "ENG",
"VendorTaxJurisdiction": "25002303",
"Status": "Available",
"VendorTaxId": "00-01101010",
"VendorAddresses": [
```

```
    {
    "VendorAddressType": "PRIMARY",
    "Address1": "123 Cedar Street",
    "Address2": "Suite 200",
    "ZipCode": "06405",
    "City": " Branford",
    "RegionISOCode": "US-CT",
    "CountryISOCode": "US",
    "TelephoneNumber": "999-99-9999",
    "FaxNumber": "999-99-9999"
}
```


Verification

Import the vendor list into InEight cloud platform using API Upsert Vendors, or manually create Vendors in the Resource Library UI.



Vendors view in resource library

Unions

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Potentially 100s of records
APIM Name		Upsert Unions
InEight Application	**Starting Version**	18.2
	Ending Version	

Fields

Depth	Name	Type	Precision	Parent	Req.
1	UnionDisplayId	String	50		Yes
1	UnionDescription	String	250		Yes
1	IsActive	Boolean	NA		No
1	SourceSystemId[1]	String	50		Yes
1	SourceSystemName	String	50		Yes

1 - Natural Key field.

Field Descriptions

Name	Description	Example
UnionDisplayId	Unique identifier that is publicly recognizable. This field shows in normal use of InEight applications as the Union ID field in the UI. The value sent in this field can match SourceSystemId if there is no need to use a public vs private unique identifier.	IWL21
UnionDescription	User friendly description of the record that will be shown in normal use of InEight applications to assist users when performing searches or selections. This field supports a large number of text characters, but it is recommended to keep the description as short as possible while retaining an easily recognized meaning.	Iron Workers Local 21
IsActive	Determines if the record is active and available to the system. If a value is not provided, the value *true* is used. Sending a value of *false* will cause a soft delete of the record in InEight.	true

Name	Description	Example
SourceSystemId	Identifier created by the "system of record" and used in all system communications as the primary method of specifying a unique record. This field does not display in normal use of InEight applications and can contain internal identifiers such as a GUID that is not understandable by typical system users.	IWL21
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	JDE

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • UnionDisplayId • UnionDescription	400	The request is invalid. The [Field Name] Field is required.
API Validation	Payload exceeds String Length Size. [Field Name] options are: • UnionDisplayId • UnionDescription • SourceSystemId • SourceSystemName	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length]. Message: The request is invalid.
Entity Logic Validation	Payload has more than one record with same SourceSystemId / DisplayId	200	"Duplicate DisplayId/SourceSystemId found. Aborting the process."
Entity Logic Validation	Duplicate source systemId/DisplayId is found in the db.	200	"Duplicate found in SourceSystemId or DisplayId for '{Display}'"
Entity Logic Validation	Payload has inactive Union	200	"Ignoring inactive Union having DisplayId '{DisplayId}' for adding"
Entity Logic Validation	Payload has Dependency	200	"Ignoring Union having SourceSystemId '{DisplayId}' for deleting due to its dependencies"

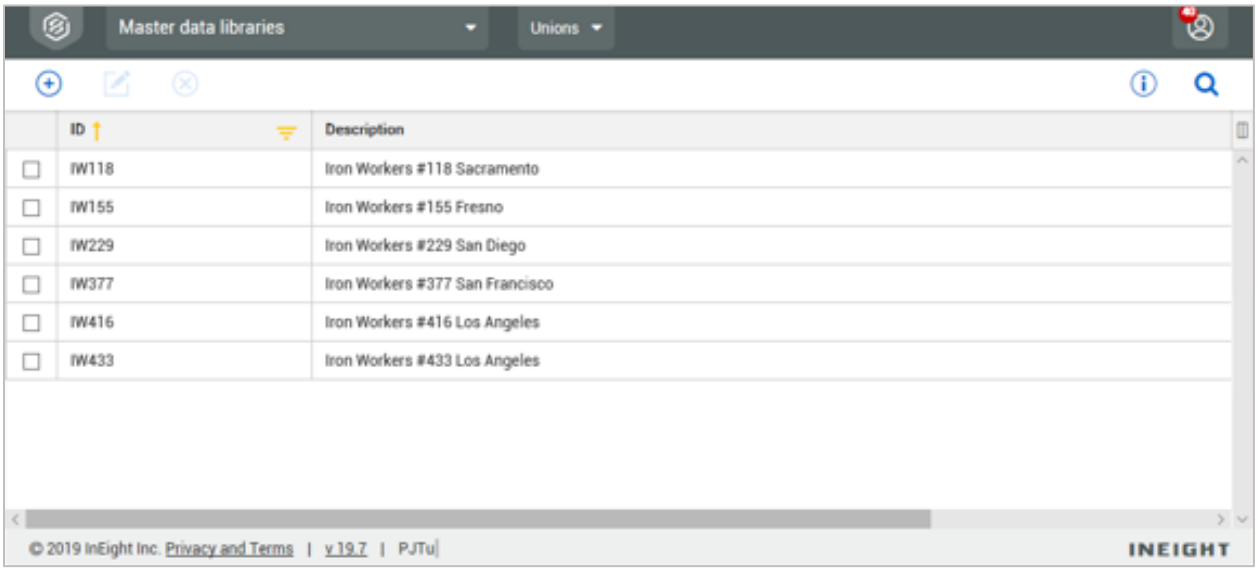
API/Entity Logic	Condition	Code	Message
Entity Logic Validation	Payload Data has Invalid Union	200	"Ignoring invalid Union having DisplayId '{DisplayId}' for update"
Entity Logic Validation	Valid Payload	200	"Imported {resultsCount} of {itemsCount} Union entities: Adds({addsCount}) Updates({updatesCount}) Deletes({deletesCount}) Errors(0) Skipped({itemsCount - resultsCount})"

Sample JSON

```
[
 {
 "UnionDisplayId": "0UFLD",
 "UnionDescription": "UnionDesc",
 "IsActive": true,
 "SourceSystemId": "0UFLD",
 "SourceSystemName": "JDE"
 },
 {
 "UnionDisplayId": "7208BC",
 "UnionDescription": "UnionDesc1",
 "IsActive": true,
 "SourceSystemId": "7208BC",
 "SourceSystemName": "JDE"
 }
]
```

Verification

Results can be viewed in the Unions screen under the menu item ‘Master data libraries’ > ‘Unions’.



Union Payroll Indicators

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Tens of records on upsert/import. Changes will be very infrequent.
APIM Name		Upsert UnionPayrollIndicators
InEight Application	**Starting Version**	18.2
	Ending Version	

Fields

Depth	Name	Type	Precision	Parent	Required
1	UnionCode	String	50		See NOTE
1	EmployeeId	String	50		See NOTE
1	PayrollIdentifier[1]	String	50		Yes
1	StartDate[2]	String	34		Yes
1	EndDate[2]	String	34		Yes

1 - Natural Key field.
2 - The data format for Date/Time fields is YYYY-MM-DDTHH:MM:SS±hhmm, where hhmm is the time zone offset. If the time is already converted to UTC, then the offset will be +0000.

NOTE:	Either UnionCode or EmployeeId is required in the data payload, however, including both at the same time for any individual record is not required.

Field Descriptions

Name	Description	Example
UnionCode	Unique identifier created by the external system of the Union to which the Payroll Indictor is being associated. Must match a valid Union SourceSystemId.	006
EmployeeId	Providing a unique employee identifier in this field will restrict the Payroll indicator to only this employee and any other employees explicitly sent in the integration as separate records with the same Payroll Indicator and Union Code. Must match a valid Employee SourceSystemId.	00165836
PayrollIdentifier	Value sent in this field must match one of the defined Attendance type or Extra pay (Allowances) records in the Progress application.	1621

Name	Description	Example
StartDate	Date that the association becomes valid and can be used within InEight Plan.	2017-12-27T05:12:35.6081608Z
EndDate	Date that the association becomes invalid and can no longer be selected within InEight Plan.	2017-12-30T05:12:35.6081608Z

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	All received records have been validated.	202	Successfully saved valid UnionPayrollIndicators.
API Validation	UnionCode is either not present or have an invalid value in a record.	200	Following UnionCodes:{comma separated list of unioncodes} does not exist in the Application.
API Validation	PayrollIdentifier is either not present or have an invalid value in a record.	200	Following PayrollIndicatorCodes:{comma separated list of PayrollIdentifiers} does not exist in the Application.
API Validation	EmployeeId is either not present or have an invalid value in a record.	200	Following Employees:{comma separated list of EmployeeIds} does not exist in the Application.

Sample JSON


```
[
{
   "UnionCode": "006",
   "EmployeeId": "00165836",
   "PayrollIdentifier": "1621",
   "StartDate": "2017-12-27T05:12:35.6081608Z",
   "EndDate": "2017-12-30T05:12:35.6081608Z"
},
{
   "UnionCode": "bulk1",
   "EmployeeId": "00210108",
   "PayrollIdentifier": "5005",
   "StartDate": "2017-12-27T05:12:35.6081608Z",
   "EndDate": "2017-12-30T05:12:35.6081608Z"
},
{
   "UnionCode": "SSId2",
   "EmployeeId": "00301410",
```

```
    "PayrollIdentifier": "4110",
    "StartDate": "2017-12-27T05:12:35.6081608Z",
    "EndDate": "2017-12-30T05:12:35.6081608Z"
}
]
```


NOTE:	As of release 20.5, StartDate and EndDate fields have been changed from datetimeoffset to date. The field still accepts datetimeoffset in the payload, but the application uses only the date portion of it.

Employees (Post)

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Entire entity – Thousands of records. Updates to records can occur on a daily basis and can affect multiple records.
APIM Name		Upsert Employees
InEight Application	**Starting Version**	18.2
	Ending Version	

Fields

Depth	Name	Type	Precision[1]	Parent	Req.
1	CraftId	String	50		Yes
1	VendorCode	String	250		No
1	EmployeeId[2]	String	50		Yes
1	FirstName	String	100		Yes
1	LastName	String	100		Yes
1	IsActive	Boolean	NA		No
1	EmployeePIN	String	64		No
1	CanReportTime	Boolean	NA		Yes
1	SourceSystemId	String	50		No
1	SourceSystemName	String	50		No
1	StartDate[3]	String	34		No
1	EndDate[3]	String	34		No
1	CurrencyCode	String	50		Yes
1	UoMName	String	50		Yes
1	StraightTimeRate	Number	16,5		No
1	OverTimeFactor	Number	16,5		No
1	OverTimeRate	Number	16,5		No
1	DoubleTimeFactor	Number	16,5		No
1	DoubleTimeRate	Number	16,5		No
1	UseBaseWageFactors	Boolean	NA		No

Depth	Name	Type	Precision[1]	Parent	Req.
1	UnionSourceSystemId	String	50		No
1	PrimaryCompany	String	100		No
1	JobTitle	String	100		No
1	UnionJobTitle	String	100		No
1	SupervisorId	String	200		No
1	CountryISOCode	String	10		No
1	DefaultCostCenterSourceSystemId	String	50		No
1	AdditionalPayrollInstructionCode	String	100		No
1	IsOverrideAdditionalPayrollInstructionCodeAllowed	Boolean	NA		No
1	IsAvailableWeeklyTimesheet	Boolean	NA		No

1 - For numeric data types, precision is given as total digits allowed in the field and the number of those digits that exist to the right of the decimal. For example, 16,5 represents a total of 16-digits allowed in the field with 5 of those digits existing as decimal places and 11 digits on the left of the decimal. The decimal is not counted as a digit.

2 - Natural Key field.

3 - The data format for Date/Time fields is YYYY-MM-DDTHH:MM:SS±hhmm, where hhmm is the time zone offset. If the time is already converted to UTC, then the offset will be +0000.

Field Descriptions

Name	Description	Example
CraftId	Identifies the craft of the employee. The value in this field must match a valid public identifier for a craft from the Crafts Integration or values manually entered in InEight.	STST
VendorCode	Identifies a vendor that exists in InEight. Employees with a VendorCode are managed by the associated vendor and their availability is limited to that vendor.	BASE
EmployeeId	Public unique Identification of the employee within InEight.	00001613
FirstName	First name of the employee.	John
LastName	Last name of the employee.	Smith
IsActive	This field is not recommended for use. Sending a value of false within this field will cause the record to be soft deleted from InEight and unrecoverable without assistance from InEight Professional Services.	true

Name	Description	Example
EmployeePIN	A PIN that is used by employees to verify themselves when approving hours on a time card. This is optional functionality in InEight. Data must be sent as numbers, and it is up to sender to zero pad the number if necessary. This is often populated with the employees date of birth, so the employee knows their unique pin number.	003456
CanReportTime	Determines if the employee can report time on a time card in InEight.	true
SourceSystemId	Identification of the employee record in the external system of records for employees. This field is used in the integration flow to perform a lookup of the employee within InEight to be updated or added. This field is also used to match employee records to other entities in integrations (e.g. ProjectEmployee) where an employee is cross-referenced.	00001613
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	JDE
StartDate	Start or hire date of the employee in the customer's organization. This is not a specific date for a project. If not provided, the system will use a default date of when the record was received.	1997-06-01T00:00:00+0000
EndDate	End or termination date of the employee from the customer's organization. This is not a specific date for a project. The date can be set to a future date as needed. If not provided, the system will use a default end date of 12/31/9999.	9999-12-31T00:00:00+0000
CurrencyCode	Identifies the currency associated with the employee which is used to determine the employee's standard rate of pay. To allow successful assignment of the employee record to a project, the value provided in this field must match one of the currencies used for projects.	USD
UoMName	Name or SourceSystemId of an existing Unit of Measure record that should be used for resources assigned to the employee. This is used to determine the employee's standard rate of pay.	Hour
StraightTimeRate	Rate for standard pay that should be applied for the employee within an assigned project. If a value of "" is sent in the integration, the system shall inherit any straight time rate assigned to the employee through their associated Craft.	35.00
OverTimeFactor	Factor used to multiply the standard rate of the employee to determine their overtime rate. (Straight Time X Overtime Factor = Overtime) Only provide this if not using OverTimeRate.	1

Name	Description	Example
OverTimeRate	Exact calculated rate paid to the employee for overtime work as per applicable laws. Only provide this if not using OverTimeFactor.	35.00
DoubleTimeFactor	Factor used to multiply the standard rate of the employee to determine their double time rate. (Straight Time X Double Time Factor = Double time) Only provide this if not using DoubleTimeRate.	2
DoubleTimeRate	Exact calculated rate paid to the employee for double time work as per applicable laws. Only provide this if not using DoubleTimeFactor.	70.00
UseBaseWageFactors	Flag that determines whether to use OverTimeRate and DoubleTimeRate or OverTimeFactor and DoubleTimeFactoror. Setting the value to true will indicate that factors should be used instead of specific rates.	false
UnionSourceSystemId	SourceSystemId of a Union to which the employee belongs. Must match a valid union from Master Data in InEight.	ILW
PrimaryCompany	Free-text field. The value of this field is used for information about the employee in InEight Time Center.	PK
JobTitle	Used to display the job title for the employee in the Daily Plan UI. If this field is blank, the UI will use the employee's assigned craft description.	Carpenter
UnionJobTitle	InEight Plan application can be configured to display the UnionJobTitle, JobTitle, or the craft associated to the Employee.	ILW Carpenter
SupervisorId	Used for reporting Staff Weekly Timesheets. A report can be sent to the User provided in this field for each associated employee that has WeeklyTimesheet set to *True*. The value of this field must match the SourceSystemId of a valid InEight user	john.smith@xyzcorp.com
CountryISOCode	Location/country of employment of the employee.	USA
DefaultCostCenterSourceSystemId	Default Cost Center for charging pay when hours should not be charged against the project to which they are assigned (e.g. special types of leave).	Cost Center 1
AdditionalPayrollInstructionCode	Default payroll processing code that can be overridden for each Time Card.	NIGHT
IsOverrideAdditionalPayroll InstructionCodeAllowed	Determines if the Additional Payroll Instruction Code can be overridden for this employee.	true
IsAvailableWeeklyTimesheet	Determines if the employee should be available for selection in a Weekly Timesheet.	false

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	All received records have been validated.	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • CraftId • EmployeeId • FirstName • LastName • CanReportTime • CurrencyCode • UoMName	400	The request is invalid. The [Field Name] Field is required.
API Validation	Payload exceeds String Length Size. [Field Name] options are: • CraftId • VendorCode • EmployeeId • FirstName • LastName • EmployeePIN • SourceSystemId • SourceSystemName • StartDate • EndDate • CurrencyCode • UoMName • UnionSourceSystemId • PrimaryCompany • JobTitle • UnionJobTitle • SupervisorId • CountryISOCode • DefaultCostCenter SourceSystemId • AdditionalPayroll InstructionCode	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length]. Message: The request is invalid.
Entity Logic Validation	Payload has more than one record with same SourceSystemId/DisplayId	200	Duplicate SourceSystemId/DisplayId {Comma separated list} found. Aborting the process.
Entity Logic Validation	Payload Data has SourceSystemId or DisplayId which already exists in database	200	Ignoring Employee '{EmployeeId}' as it is having duplicate SourceSystemId or DisplayId
Entity Logic Validation	Payload has invalid CraftId	200	Ignoring Employee '{EmployeeId}' as it is having invalid CraftId {CraftSourceSystemId}'

API/Entity Logic	Condition	Code	Message
Entity Logic Validation	Payload has invalid CurrencyCode.	200	Ignoring Employee '{EmployeeId}' as it is having invalid CurrencyCode {CurrencyCode}'
Entity Logic Validation	Payload has invalid UOMName.	200	Ignoring Employee '{EmployeeId}' as it is having invalid UOMName/UOMSourceSystemId '{UOMName}'
Entity Logic Validation	Payload has invalid VendorSourceSystemId.	200	Ignoring Employee '{EmployeeId}' as it is having invalid VendorCode '{VendorSourceSystemId}'
Entity Logic Validation	Payload has invalid UnionSourceSystemId.	200	Ignoring Employee '{EmployeeId}' as it is having invalid UnionSourceSystemId '{UnionSourceSystemId}'
Entity Logic Validation	Payload has invalid SupervisorSourceSystemId.	200	Ignoring Employee '{EmployeeId}' as it is having invalid SupervisorId '{Supervisor.SourceSystemId}'
Entity Logic Validation	Payload has invalid CountryISOCode.	200	Ignoring Employee '{EmployeeId}' as it is having invalid CountryISOCode '{CountryISOCode}'
Entity Logic Validation	Payload has invalid CostCenterSourceSystemId.	200	Ignoring Employee '{EmployeeId}' as it is having invalid DefaultCostCenterSourceSystemId '{CostCenter.SourceSystemId}'
Entity Logic Validation	Payload VendorId of the Employee not equal to VendorId of the Craft Employee	200	Ignoring Employee '{EmployeeId}' because No crafts available for this vendor and union selection.
Entity Logic Validation	Payload UnionID of the Employee not equal to UnionID of the Craft Employee	200	Ignoring Employee '{EmployeeId}' because No crafts available for this vendor and union selection.
Entity Logic Validation	Payload Employee.StartDate is less than 1900/1/1	200	Ignoring employee having employee display '{EmployeeId}' as {employee} start date should be after 01/01/1900
Entity Logic Validation	Payload Employee.StartDate is greater than Employee.EndDate	200	Ignoring Employee '{EmployeeId}' due to End date cannot occur before Start date
Entity Logic Validation	Payload has inactive records while inserting	200	Ignoring inactive Employee '{EmployeeId}' for adding
Entity Logic Validation	Payload has any rates/factors without ST rate	200	Ignoring Employee '{EmployeeId}' because ST rate is not passed
Entity Logic Validation	Payload has any rates/factors with a negative value.	200	Ignoring Employee '{EmployeeId}' because Rates/Factors should not accept -ve value
Entity Logic Validation	Payload has employee as Supervisor of themself.	200	Ignoring Employee '{EmployeeId}' for updating because Employee cannot be a Supervisor of itself
Entity Logic Validation	Payload has employee with a currency that is not part of project currencies of all the projects the employee is assigned to.	200	Ignoring Employee '{EmployeeId}' for updating because User cannot change the currency of an employee to other currency that is not part of project currencies of all the projects he/she is assigned to.

API/Entity Logic	Condition	Code	Message
Entity Logic Validation	Payload has employee with a different vendor than that of the Project craft (Rate code).	200	Ignoring Employee '{EmployeeId}' for updating because Project craft is specific to a vendor and can not be assigned to an employee of a different vendor

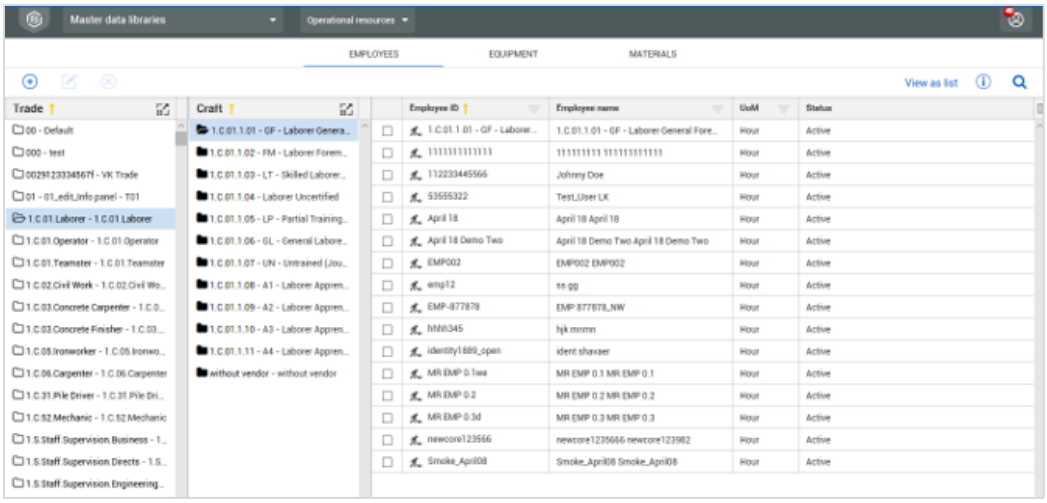
Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

Sample JSON


```
{
 "CraftId": "STST",
 "VendorCode": "BASE",
 "EmployeeId": "00001613",
 "FirstName": "John",
 "LastName": "Smith",
 "IsActive": true,
 "EmployeePIN": "003456",
 "CanReportTime": true,
 "SourceSystemId": "00001613",
 "SourceSystemName": "CLNTERPSOU",
 "StartDate": "1997-06-01T00:00:00+0000",
 "EndDate": "9999-12-31T00:00:00+0000",
 "CurrencyCode": "USD"
 "UOMName": "Hour",
 "StraightTimeRate": 35.00,
 "OverTimeFactor": 1,
 "OverTimeRate": 35.00,
 "DoubleTimeFactor": 2,
 "DoubleTimeRate": 70.00,
 "UseBaseWageFactors": true,
 "UnionSourceSystemId": "ILW",
 "PrimaryCompany": "PK",
 "JobTitle": "Carpenter",
 "UnionJobTitle": "ILW Carpenter",
 "SupervisorId": "john.smith@xyzcorp.com",
 "CountryISOCode": "USA",
 "DefaultCostCenterSourceSystemId": "Cost Center 1",
 "AdditionalPayrollInstructionCode": "NIGHT",
 "IsOverrideAdditionalPayrollInstructionCodeAllowed": false,
 "IsAvailableWeeklyTimeSheet": true
}
```


Verification

To view Employees, go to Master data libraries > Operational resources > **EMPLOYEES** tab.


Master data libraries
Operational resources
EMPLOYEES
EQUIPMENT
MATERIALS
View as list
Trade
Craft
Employee ID
Employee name
UoM
Status

Employees (Patch)

This integration should be used only to solve specific data maintenance issues related to having multiple sources of entry (e.g. partially in an HCM system and partially in InEight).

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		Patch Employees
InEight Application	**Starting Version**	19.7
	Ending Version	

Fields

IMPORTANT:	When using this integration, any field that is not a required field can be excluded from the JSON. If a field is included in the JSON, the InEight data will be updated even if the field is blank or contains a null value.

See Employees (Post) Fields for detailed information.

Field Descriptions

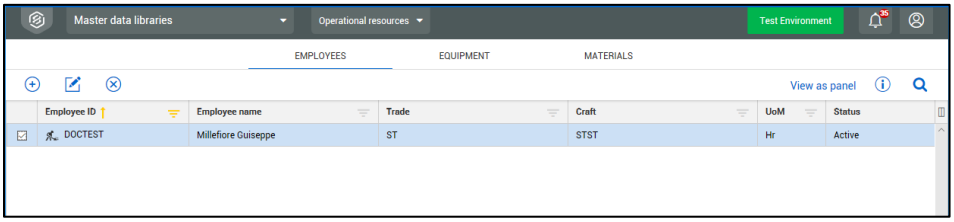
See Employees (Post) Field Descriptions for detailed information.

Sample JSON

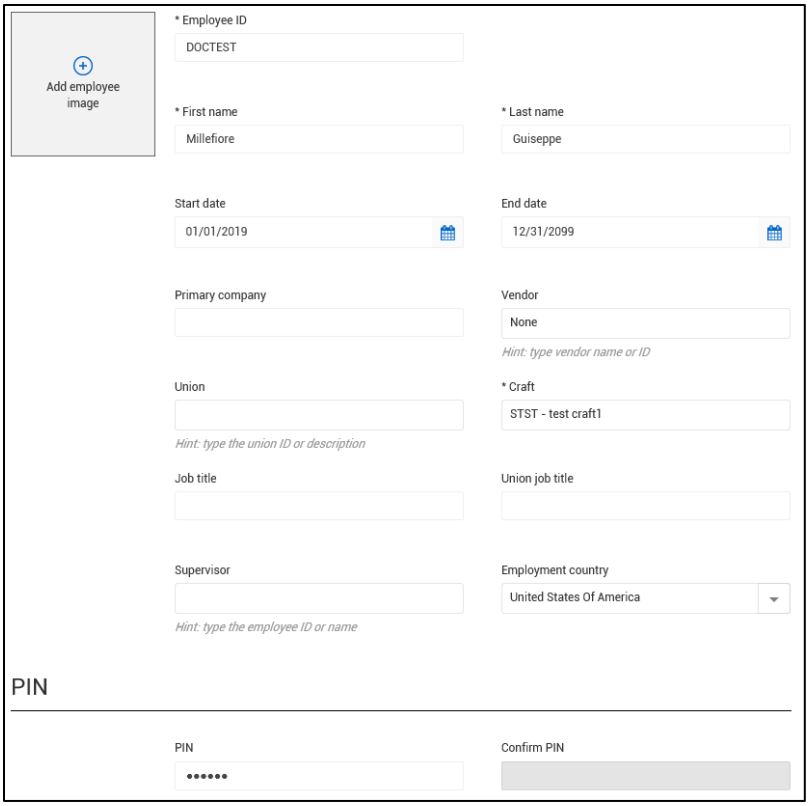
```
[
 {
 "EmployeePIN": "445566",
 "SourceSystemId": "GMILLE",
 }
]
```

Verification

To view changes to Employee records, go to Master data libraries > Operational resources > **EMPLOYEES** tab.



Select a record, and then select the **Edit** icon to open the details for an employee.



Employees (Get)

This API uses the standard InEight asynchronous GET pattern (refer to *"Getting Started with InEight Integrations – Developer Guide"*) and returns 500 records for each request. Use the OData query parameter $skip to continue to make subsequent calls until the returned record set falls below a count of 500.

Direction		From InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		All Employee records
APIM Name		List Employees
InEight Application	**Starting Version**	19.1
	Ending Version	

Supported Filters

This API supports standard OData query parameters of $expand, $filter, $select, $orderby, $top, $skip, and $count.

Fields

NOTE:	Values in the EmployeePIN field are masked for security purposes.

See Employees (Post) Fields for detailed information.

Field Descriptions

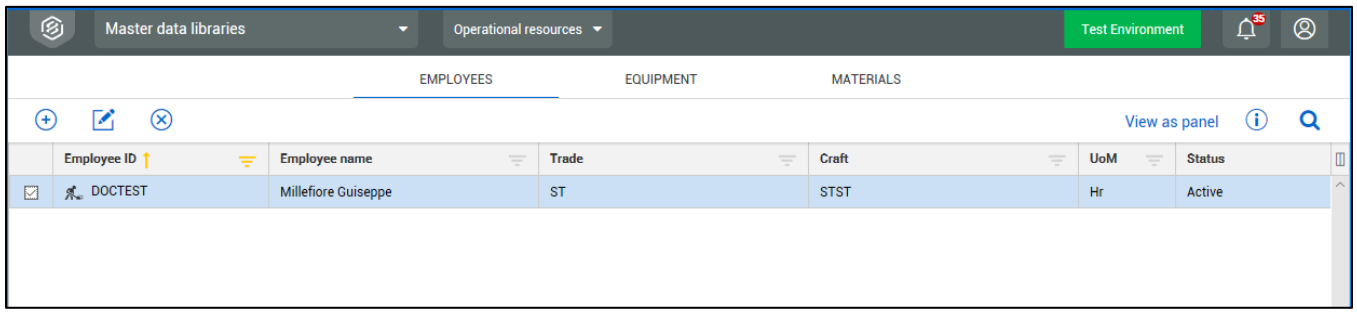
See Employees (Post) Field Descriptions for detailed information.

Sample JSON

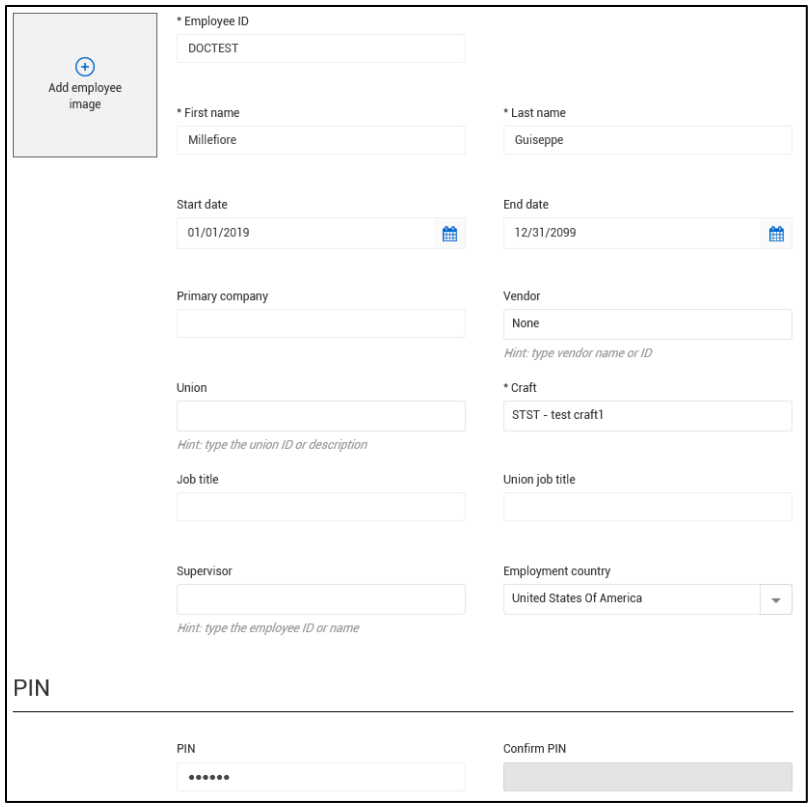

```
{"@odata.context":"https://apitest.ineight.com/integrations/$metadata#Employees","v
alue":[{"CraftId":"00","EmployeeId":"00165836","FirstName":"Nandhu","LastName":"Sak
thi
sep","SourceSystemId":"00165836","VendorCode":"","IsActive":true,"EmployeePIN":"824
CFF6EE6A62C7E2E4D26716C4A90F53637DFD4538D9F8A2550C527275A250C","CanReportTime":fals
e,"SourceSystemName":"Nnadysep18th","StartDate":"2005-06-
01T00:00:00+00:00","EndDate":"2018-01-
19T00:00:00+00:00","CurrencyCode":"USD","UOMName":"Hour","StraightTimeRate":15.0000
0,"OverTimeFactor":1.44333,"OverTimeRate":21.65000,"DoubleTimeFactor":0.39133,"Doub
leTimeRate":5.87000,"UseBaseWageFactors":false,"UnionSourceSystemId":"","PrimaryCom
pany":"","JobTitle":"","UnionJobTitle":"","SupervisorId":"","CountryISOCode":"","De
faultCostCenterSourceSystemId":"","AdditionalPayrollInstructionCode":"","IsOverride
AdditionalPayrollInstructionCodeAllowed":false,"IsAvailableWeeklyTimeSheet":true},{
"CraftId":"STST","EmployeeId":"00210108","FirstName":"Matthew","LastName":"Easter",
"SourceSystemId":"00210108","VendorCode":"","IsActive":true,"EmployeePIN":"4DCF7298
CB289B64240C9E76738FE8C25622E7BA268771D903872FF676FD2A71","CanReportTime":false,"So
urceSystemName":"","StartDate":"2009-11-30T00:00:00+00:00","EndDate":"9999-12-
31T00:00:00+00:00","CurrencyCode":"USD","UOMName":"Hour","StraightTimeRate":15.0000
0,"OverTimeFactor":1.44333,"OverTimeRate":21.65000,"DoubleTimeFactor":0.39133,"Doub
leTimeRate":5.87000,"UseBaseWageFactors":false,"UnionSourceSystemId":"","PrimaryCom
pany":"","JobTitle":"","UnionJobTitle":"","SupervisorId":"","CountryISOCode":"","De
faultCostCenterSourceSystemId":"","AdditionalPayrollInstructionCode":"","IsOverride
AdditionalPayrollInstructionCodeAllowed":false,"IsAvailableWeeklyTimeSheet":false},
{"CraftId":"STST","EmployeeId":"00301410","FirstName":"Padmaja","LastName":"Marupud
i","SourceSystemId":"00301410","VendorCode":"","IsActive":true,"EmployeePIN":"09B2D
FA65B2642381CD07D3DF48C4E232E9DAE9B85C54D5EB7B39D3D532E0DED","CanReportTime":true,"
SourceSystemName":"","StartDate":"2012-07-25T00:00:00+00:00","EndDate":"9999-12-
31T00:00:00+00:00","CurrencyCode":"PLN","UOMName":"Hour","StraightTimeRate":2.00000
,"OverTimeFactor":"","OverTimeRate":0.00000,"DoubleTimeFactor":"","DoubleTimeRate":
0.00000,"UseBaseWageFactors":false,"UnionSourceSystemId":"","PrimaryCompany":"","Jo
bTitle":"","UnionJobTitle":"","SupervisorId":"","CountryISOCode":"","DefaultCostCen
terSourceSystemId":"","AdditionalPayrollInstructionCode":"","IsOverrideAdditionalPa
yrollInstructionCodeAllowed":false,"IsAvailableWeeklyTimeSheet":false}]}
```

Verification

To view Employee records, go to Master data libraries > Operational resources > **EMPLOYEES** tab.



Select a record, and then select the **Edit** icon to open the details for an employee.



Contacts

Contacts (Import)

New contacts and updates to account codes performed using this API are automatically published (instead of being staged) and are immediately available for use in InEight cloud applications.

IMPORTANT:	When using this integration, any field that is not a required field can be excluded from the JSON. If a field is included in the JSON, the InEight data will be updated even if the field is blank or contains a null value.

Direction		From external system to InEight cloud platform.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		Upsert Contacts
InEight Application	**Starting Version**	20.11
	Ending Version	

Contacts (Get)

In addition to reading and exporting information for a master list of contacts, a customer can use this integration to query contacts using SSID, name, company, or other contact details.

Direction		From InEight cloud platform to external system.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		List Contacts
InEight Application	**Starting Version**	20.11
	Ending Version	

Fields

Depth	Name	Type	Precision[1]	Parent	Req.
1	Firstname	String	100		Yes
1	Lastname	String	100		Yes
1	Company	String	200		Yes
1	PrimaryEmail	String	200		Yes
1	SecondaryEmail	String	200		Yes
1	SourceSystemId	Boolean	200		Yes
1	SourceSystemName	String	50		Yes
1	ContactTypeReference	String	50		Yes
1	LanguagePreference	String	5		No
1	VendorSourceSystemId	String	250		No
1	Department	String	250		No
1	Title	String	250		No
1	Position	String	250		No
1	OfficeNumber	String	50		No
1	MobileNumber	String	50		No
1	FaxNumber	String	50		No
1	Address1	String	250		No
1	Address2	String	250		No
1	City	String	100		No
1	Zipcode	String	50		No
1	RegionISOCode	String	10		No
1	CountryISOCode	String	10		No
1	IsActive	Boolean	NA		No

Field Descriptions

Name	Description	Example
Firstname	First name of the contact.	John
Lastname	Last name of the contact.	Smith
Company	Name of the company that the contact works for.	Acme Construction
PrimaryEmail	Primary email address for the contact.	john.smith@acmeconstruction.com

Name	Description	Example
SecondaryEmail	Secondary email address for the contact.	John.smith@gmail.com
SourceSystemId	External unique identifier of the contact record provided by the system of record (SOR) for the contact.	d5a9936eb24d474d999667e76bdd8896
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	SAP-HR
ContactTypeReference	Category for the contact.	Vendor
LanguagePreference	Preferred language of conversation of the contact.	English
VendorSourceSystemId	If the contact is associated to a vendor, this is the external identifier (SourceSystemId field) from the vendor's system of record (SOR).	09690e390f804eb4ab070f8219d70937
Department	Department that the contact belongs to or has been assigned temporarily, in their company.	Marketing
Title	Job title of the contact in their company.	Director
Position	Job position of the contact in their company.	Director
OfficeNumber	Office phone number for the contact.	123-456-7890
MobileNumber	Mobile phone number for the contact.	123-456-7890
FaxNumber	Fax number for the contact.	123-456-7890
Address1	Street address for the contact (line 1).	231 Elm Street
Address2	Street address for the contact (line 2).	
City	City of residence or work for the contact.	Menlo Park
Zipcode	Postal code for the contact.	99867
RegionISOCode	ISO region code (state) where the contact lives/works. **NOTE:** Make sure to enter US-<*state abbreviation*> in this field. If you enter only the state abbreviation, the payload will fail.	US-CA
CountryISOCode	ISO country code where the contact lives/works.	US
IsActive	Defines if the contact is an active employee of their company.	true

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • FirstName • LastName • Company • PrimaryEmail • SourceSystemId • ContactTypeReference	400	The request is invalid. The [Field Name] Field is required.
API Validation	Payload exceeds String Length Size.	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length].
API Validation	Payload data is Empty	400	Payload was missing or invalid and was not parsed.
Entity Logic Validation	Payload has more than one record with same SourceSystemId	200	Duplicate SourceSystemId {Comma separated list of SourceSystemIds} found. Aborting the process.
Entity Logic Validation	Payload has more than one record with same DisplayId	200	Duplicate DisplayId {Comma separated list of PrimaryEmail} found. Aborting the process.
Entity Logic Validation	Payload has conflict with uniqueness on SourceSystemId or DisplayId with existing data	200	Ignoring Contact '{PrimaryEmail}' as it is having duplicate SourceSystemId or DisplayId
Entity Logic Validation	Payload has inactive records while inserting	200	Ignoring inactive Contact '{PrimaryEmail}' for adding

Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

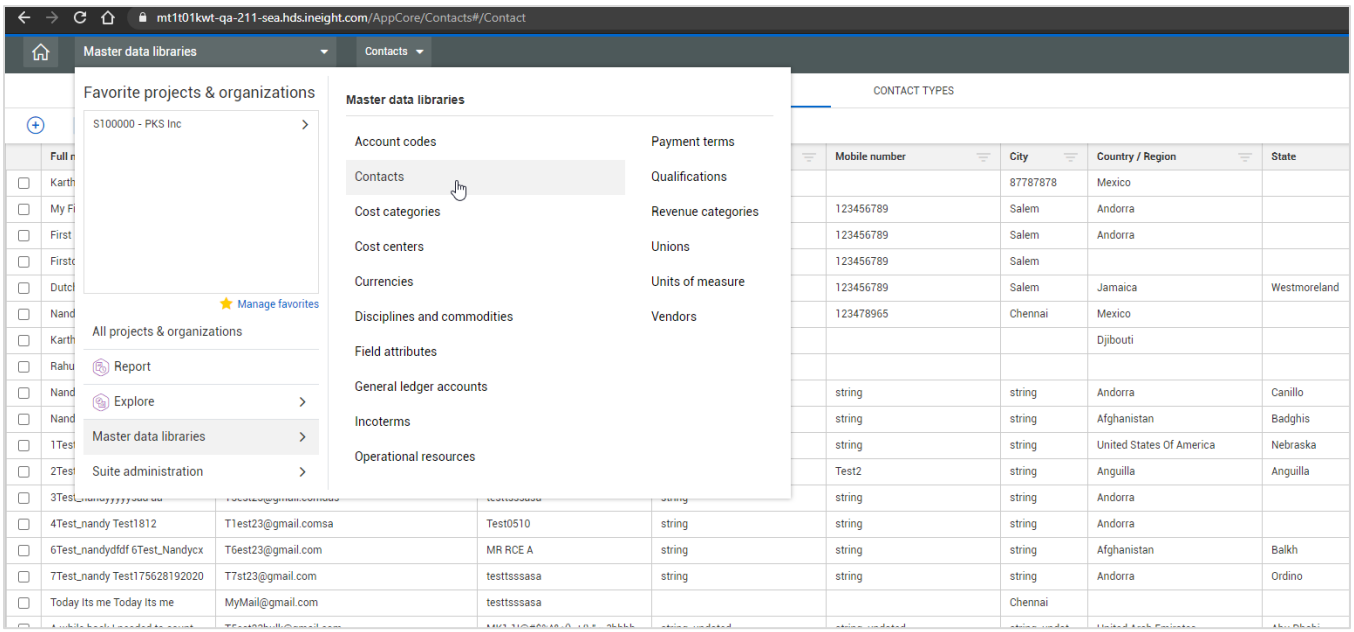
Sample JSON


```
[
   {
   "FirstName": "John",
   "LastName": "Smith",
   "Company": "InEight",
   "PrimaryEmail": "satish.iyer@ineight.com",
   "SecondaryEmail": "iyer_satish_r@yahoo.com",
   "SourceSystemId": "satish.iyer@ineight.com",
   "SourceSystemName": "",
   "ContactTypeReference": "Vendor",
   "LanguagePreference": "",
   "VendorSourceSystemId": "",
   "Department": "Product",
   "Title": "P.O",
   "Position": "P.O",
   "OfficeNumber": "4024406606",
   "MobileNumber": "6145781277",
   "FaxNumber": "4024406606",
```

```
    "Address1": "123 Oak Street",
    "Address2": "Suite 12",
    "City": "Bankgor",
    "ZipCode": "100067",
    "RegionISOCode": "US-ME",
    "CountryISOCode": "US",
    "IsActive": true
    }
]
```

Verification

Verify whenever a contact is added or updated in the InEight cloud platform via the UI, an external system, or APIM. From the main menu, go to Master data libraries > **Contacts** and search for the contact to validate that it was imported correctly.



Markets

Markets (Import)

New markets and updates to markets performed using this API are automatically published (instead of being staged) and are immediately available for use in InEight cloud applications.

IMPORTANT:	When using the import integration, any field that is not a required field can be excluded from the JSON. If a field is included in the JSON, the InEight data will be updated even if the field is blank or contains a null value.

Direction		From external system to InEight cloud platform.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		Upsert Markets
InEight Application	**Starting Version**	21.3
	Ending Version	

Markets (Get)

In addition to reading and exporting information for a master list of markets, a customer can use this integration to query markets using SSID and market name.

Direction		From InEight cloud platform to external system.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		List Markets
InEight Application	**Starting Version**	21.3
	Ending Version	

Fields

Depth	Name	Type	Precision	Parent	Req.
1	Name	String	100		Yes
1	Description	String	100		Yes
1	SourceSystemId	String	200		Yes
1	SourceSystemName	String	200		Yes
1	IsActive	String	200		Yes

Field Descriptions

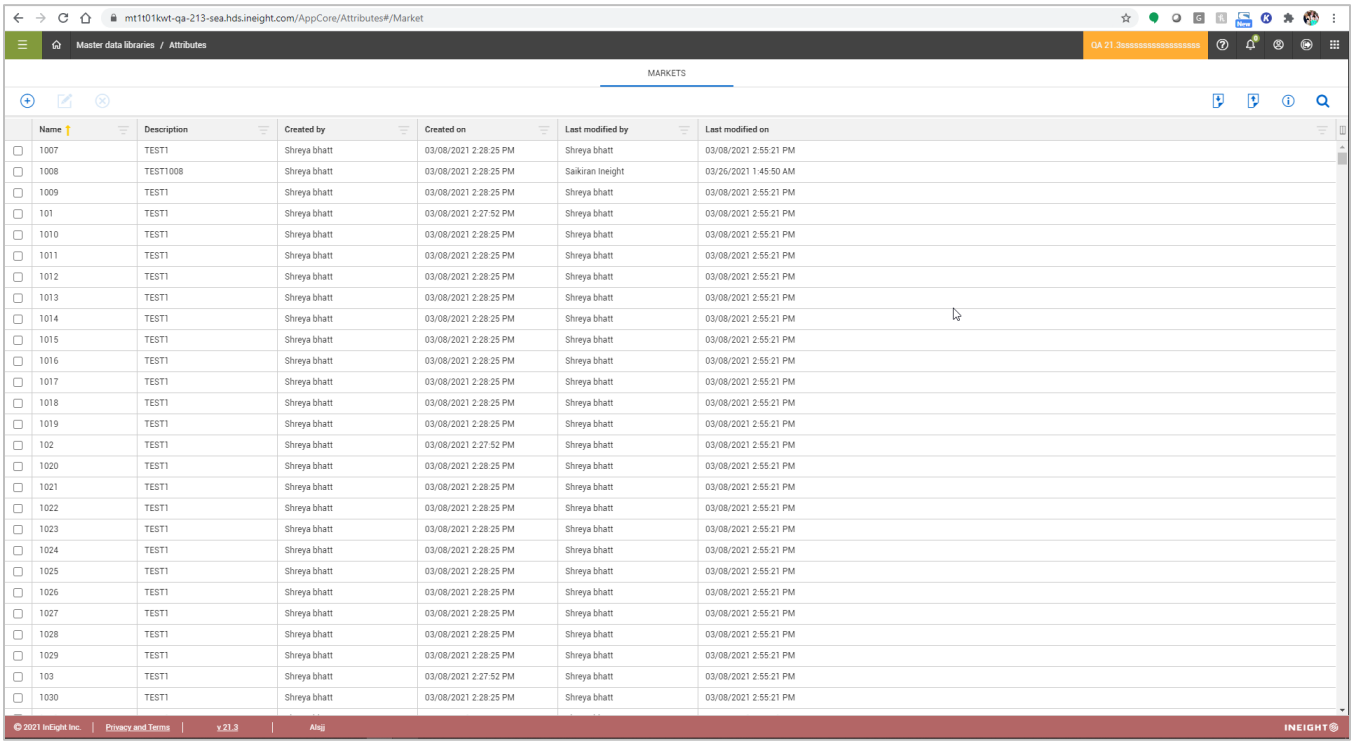
Name	Description	Example
Name	Name of the market.	Mining
Description	Description of the market.	Iron Ore mining in Utah
SourceSystemId	External identifier of the market record provided by the system of record for markets.	Satish.iyer@ineight.com
SourceSystemName	Customer provided name of the system of record for markets.	Billing
IsActive	Determines if the record is available for use.	True

Sample JSON

```
[
   {
   "Name": "John",
   "Description": "Smith",
   "SourceSystemID": "john.smithInEight.com",
   "SourceSystemName": "Billing",
   "IsActive": "T",
   }
]
```

Verification

Verify whenever a market is added or updated in the InEight cloud platform via the UI, an external system, or APIM. From the main menu, go to Master data libraries > Attributes > **Markets** and search for the market to validate that it was imported correctly.



Currencies

Currencies (Get)

This integration allows customers to query, export, and report on their master list of currencies that can be used for various business functions in multiple InEight cloud platform applications.

Direction		From InEight cloud platform to external system.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		List Currencies
InEight Application	**Starting Version**	21.7
	Ending Version	

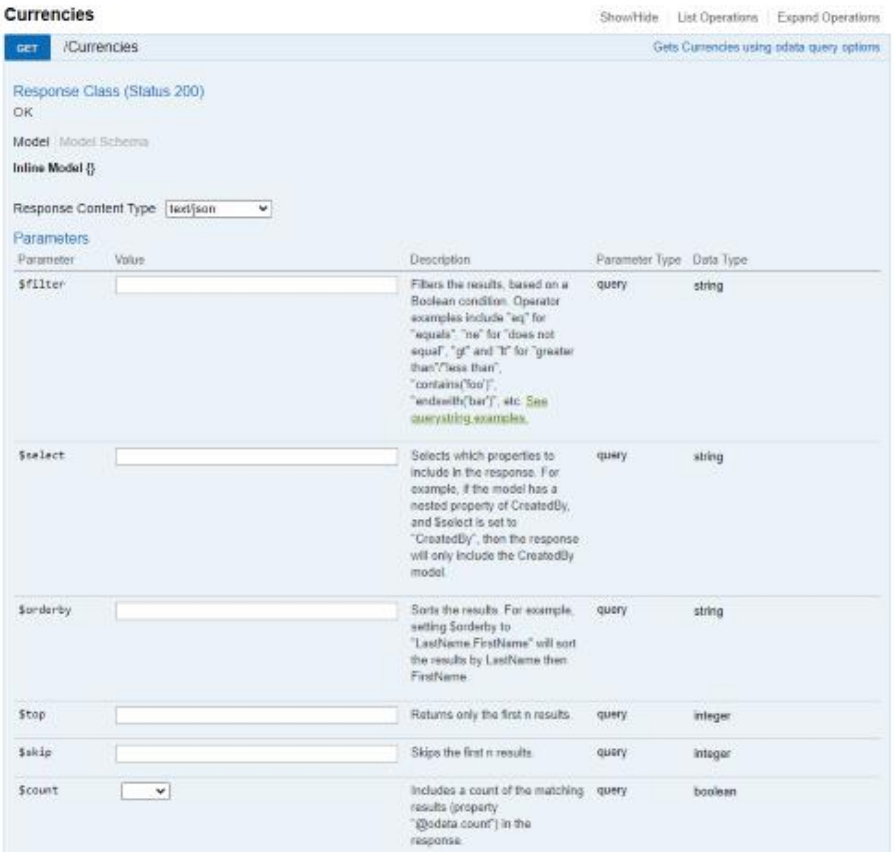
Fields

Depth	Name	Type	Precision	Parent	Req.
1	Code	String	50		Yes
1	Name	String	250		Yes
1	Entity	String	500		No
1	Symbol	String	10		No
1	CurrencySubunitName	String	250		No
1	CurrencyPluralName	String	250		No
1	CurrencySubunitPluralName	String	250		No

Field Descriptions

Name	Description	Example
Code	The public unique code that identifies the currency	EN0001
Name	Full name of the currency	U.S Dollars
Entity	Owner of the currency	USA
Symbol	Symbol of the currency	Dollar Sign
CurrencySubunitName	Full sub-unit name of the currency	Cent
CurrencyPluralName	Full plural name of the currency	Dollars
CurrencySubunitPluralName	Full plural name of the sub-unit of the currency	Cents

Sample JSON



Currencies

Show/Hide | List Operations | Expand Operations

GET /Currencies — Gets Currencies using odata query options

Response Class (Status 200)

OK

Model | Model Schema

Inline Model {}

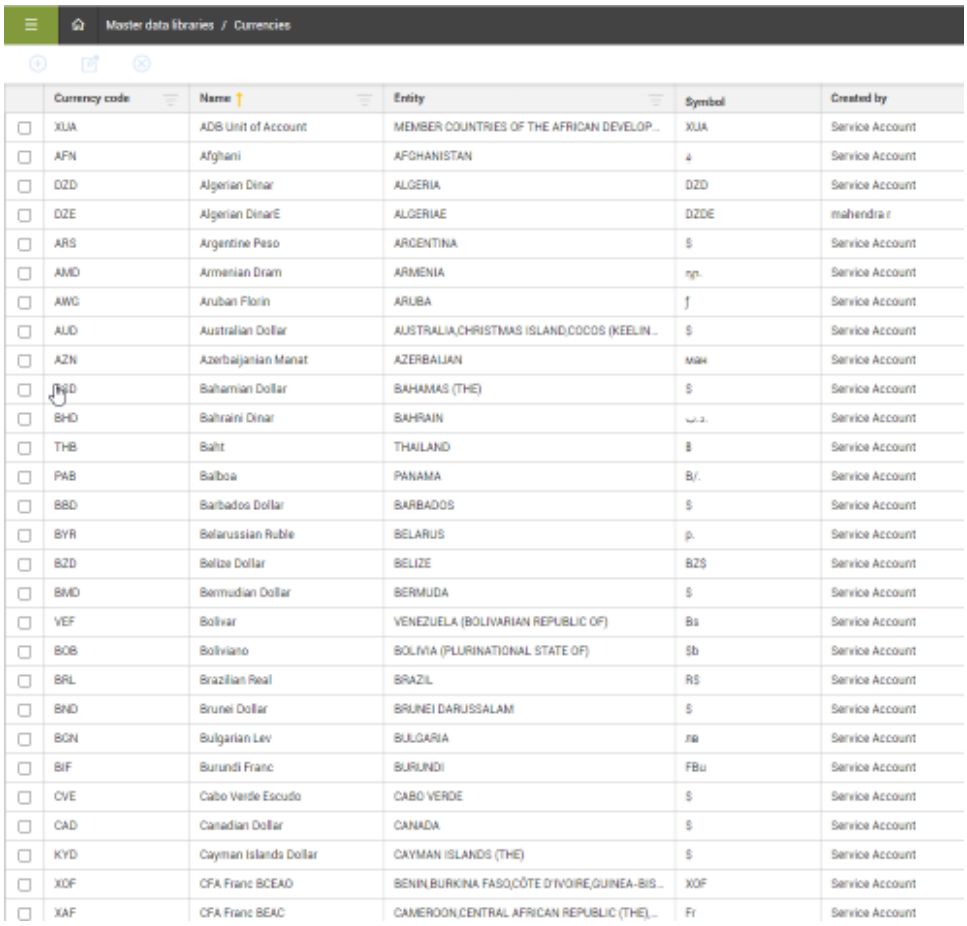
Response Content Type: text/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
$filter		Filters the results, based on a Boolean condition. Operator examples include "eq" for "equals", "ne" for "does not equal", "gt" and "lt" for "greater than"/"less than", "contains('foo')", "endswith('bar')", etc. See querystring examples.	query	string
$select		Selects which properties to include in the response. For example, if the model has a nested property of CreatedBy, and $select is set to "CreatedBy", then the response will only include the CreatedBy model.	query	string
$orderby		Sorts the results. For example, setting $orderby to "LastName,FirstName" will sort the results by LastName then FirstName.	query	string
$top		Returns only the first n results.	query	integer
$skip		Skips the first n results.	query	integer
$count		Includes a count of the matching results (property "@odata.count") in the response.	query	boolean

Verification

Currencies are master data that can be viewed and edited with required permissions and accessed in Main Menu > Master data libraries > **Currencies**.



Currency code	Name	Entity	Symbol	Created by
XUA	ADB Unit of Account	MEMBER COUNTRIES OF THE AFRICAN DEVELOP...	XUA	Service Account
AFN	Afghani	AFGHANISTAN	؋	Service Account
DZD	Algerian Dinar	ALGERIA	DZD	Service Account
DZE	Algerian DinarE	ALGERIAE	DZDE	mahendrar
ARS	Argentine Peso	ARGENTINA	$	Service Account
AMD	Armenian Dram	ARMENIA	դր.	Service Account
AWG	Aruban Florin	ARUBA	ƒ	Service Account
AUD	Australian Dollar	AUSTRALIA,CHRISTMAS ISLAND,COCOS (KEELIN...	$	Service Account
AZN	Azerbaijanian Manat	AZERBAIJAN	ман	Service Account
BSD	Bahamian Dollar	BAHAMAS (THE)	$	Service Account
BHD	Bahraini Dinar	BAHRAIN	.د.ب	Service Account
THB	Baht	THAILAND	฿	Service Account
PAB	Balboa	PANAMA	B/.	Service Account
BBD	Barbados Dollar	BARBADOS	$	Service Account
BYR	Belarussian Ruble	BELARUS	p.	Service Account
BZD	Belize Dollar	BELIZE	BZ$	Service Account
BMD	Bermudian Dollar	BERMUDA	$	Service Account
VEF	Bolivar	VENEZUELA (BOLIVARIAN REPUBLIC OF)	Bs	Service Account
BOB	Boliviano	BOLIVIA (PLURINATIONAL STATE OF)	$b	Service Account
BRL	Brazilian Real	BRAZIL	R$	Service Account
BND	Brunei Dollar	BRUNEI DARUSSALAM	$	Service Account
BGN	Bulgarian Lev	BULGARIA	лв	Service Account
BIF	Burundi Franc	BURUNDI	FBu	Service Account
CVE	Cabo Verde Escudo	CABO VERDE	$	Service Account
CAD	Canadian Dollar	CANADA	$	Service Account
KYD	Cayman Islands Dollar	CAYMAN ISLANDS (THE)	$	Service Account
XOF	CFA Franc BCEAO	BENIN,BURKINA FASO,CÔTE D'IVOIRE,GUINEA-BIS...	XOF	Service Account
XAF	CFA Franc BEAC	CAMEROON,CENTRAL AFRICAN REPUBLIC (THE),...	Fr	Service Account

Units of Measure

UOMs (Get)

This integration allows customers to query, export, and report on their master list of currencies that can be used for various business functions in multiple InEight cloud platform applications.

Direction		From InEight cloud platform to external system.
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Determined by external system.
APIM Name		List UOMs
InEight Application	**Starting Version**	19.9
	Ending Version	

Fields

Depth	Name	Type	Precision	Parent	Req.
1	Name	String	255		Yes
1	Description	String	250		Yes
1	MeasurementTypeName	String	50		Yes
1	MeasurementSyatemName	String	100		Yes
1	AlternateUOMSourceSystemID	String	200		Yes
1	ConversionFactor	String	250		No
1	BaseUOMName	String	50		No
1	DefaultAlternateMeasurement	String	50		No
1	IsBaseUOM	String	5		No
1	SKFUOM	String	250		No
1	SourceSystemID	String	250		No
1	SourceSystemName	String	250		No
1	IsActive	String	250		No

Descriptions

Name	Description	Example
Name	Name of the UOM	Time
Description	Description of the UOM	Measuring the elapsed time

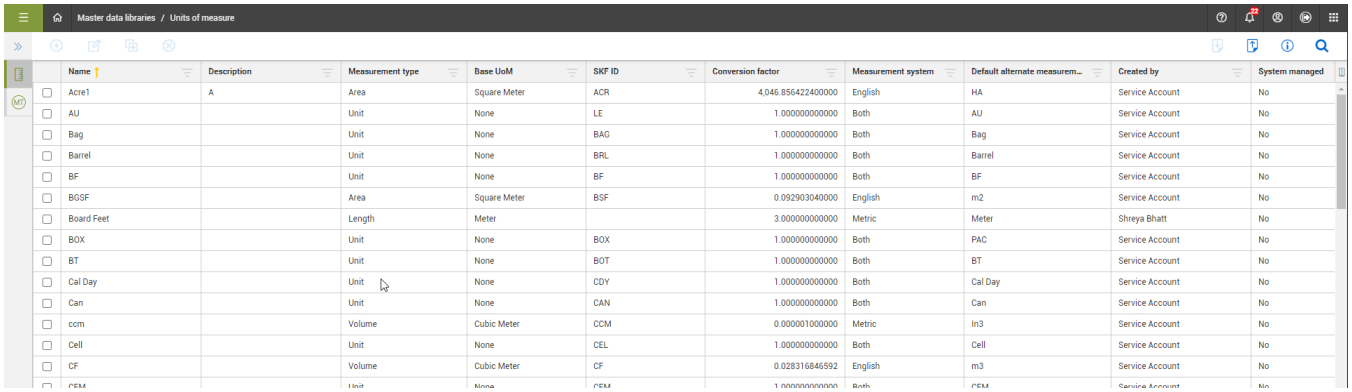
Name	Description	Example
MeasurementTypeName	Name of the Measurement Type	Seconds
MeasurementSyatemName	European or Metric System	Metric, EU
AlternateUOMSourceSystemID	System generated unique ID UOM	d5a9936eb24d474d999667e76bdd8896
ConversionFactor	The mathematical formula/algorithm needed to convert a UOM value from one system to another	Farenheit = (Celcius - 30)/2
BaseUOMName	Name of the base UOM	Time
DefaultAlternateMeasurement	Any alternate acceptable way of measureing the base UOM	Time
IsBaseUOM	Boolean value T/F	Y/N
SKFUOM	Value of the SKF UOM	Time
SourceSystemID	If UoM is entered in the UI, a GUID is system generated. If added via integration this will be the SourceSystemId from the payload.	d5a9936eb24d474d999667e76bdd8896
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	ESB
IsActive	Boolean T/F	T/F

Sample JSON

```
[
   {
   "Name": "Elapsed time",
   "Description": "Amount of time elapsed between contracts",
   "MeasurementTypeName": "Time",
   "MeasurementSystemName": "Metric",
   "AlternateUomSourceSystemId": "d5a9936eb24d47",
   "ConversionFactor": "Farenheit = (Celcius - 30)/2",
   "IsBaseUOM": true,
   "BaseUOMName": "Elapsed time",
   "SKFUOM": "Time",
   "IsActive": true,
   "SourceSystemId": "d5a9936eb24d47",
   "SourceSystemName": "ESB"
   }
]
```

Verification

UOMs are master data that can be viewed and edited with required permissions and accessed in Main Menu > Master data libraries > **Units of measure**.



Name	Description	Measurement type	Base UoM	SKF ID	Conversion factor	Measurement system	Default alternate measurem...	Created by	System managed
Acre1	A	Area	Square Meter	ACR	4,046.856422400000	English	HA	Service Account	No
AU		Unit	None	LE	1.000000000000	Both	AU	Service Account	No
Bag		Unit	None	BAG	1.000000000000	Both	Bag	Service Account	No
Barrel		Unit	None	BRL	1.000000000000	Both	Barrel	Service Account	No
BF		Unit	None	BF	1.000000000000	Both	BF	Service Account	No
BGSF		Area	Square Meter	BSF	0.092903040000	English	m2	Service Account	No
Board Feet		Length	Meter		3.000000000000	Metric	Meter	Shreya Bhatt	No
BOX		Unit	None	BOX	1.000000000000	Both	PAC	Service Account	No
BT		Unit	None	BOT	1.000000000000	Both	BT	Service Account	No
Cal Day		Unit	None	CDY	1.000000000000	Both	Cal Day	Service Account	No
Can		Unit	None	CAN	1.000000000000	Both	Can	Service Account	No
ccm		Volume	Cubic Meter	CCM	0.000001000000	Metric	In3	Service Account	No
Cell		Unit	None	CEL	1.000000000000	Both	Cell	Service Account	No
CF		Volume	Cubic Meter	CF	0.028316846592	English	m3	Service Account	No
CFM		Unit	None	CFM	1.000000000000	Both	CFM	Service Account	No

General Ledger

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Possibly hundreds of records to allow for all types of work supported by a customer.
APIM Name		Upsert GeneralLedger
InEight Application	**Starting Version**	20.11
	Ending Version	

Fields

Depth	Name	Type	Precision	Parent	Req.
1	DisplayId	String	100		Yes
1	Description	String	255		Yes
1	SourceSystemId	String	250		No
1	SourceSystemName	String	50		No
1	IsAvailable	Boolean	NA		No
1	Translations	Array			
2	Description	String	255	Translations	No
2	Language	String	5	Translations	No

Field Descriptions

Name	Description	Example
DisplayId	A code (usually 2 or 6 characters) that represents or describes the General Ledger. This is the General Ledger Accounts field in the UI.	5211012
Description	Customer-defined field that is typically used to describe or provide details about the General Ledger.	Craft Administration
SourceSystemId	The identification of the General Ledger used within the external system. In most cases, this matches the DisplayId field. This field is used by InEight to match General Ledger records to other entities where an Account Ledger is cross-referenced.	5211012
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	CLNTERPSOU

Name	Description	Example
IsAvailable	Determines the status of the account/record and whether it can be assigned. *True* equals available and *false* equals unavailable.	true
Translations	Array header for translation localized text.	
Description	Customer-defined field that is typically used to describe or provide details about the General Ledger as per the language code	Craft Administration
Language	Language code that the general ledger is shown in the UI.	EN

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • DisplayId • Description	400	The request is invalid. The [Field Name] Field is required.
API Validation	Payload exceeds String Length Size. [Field Name] options are: • DisplayId • Description • Language • SourceSystemId • SourceSystemName	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length].
Entity Logic Validation	Payload has more than one record with same SourceSystemId	200	Duplicate General ledger accounts SourceSystemId: {SourceSystemIds} found. Aborting the process.
Entity Logic Validation	Payload has more than one record with same DisplayId	200	Duplicate General ledger accounts : {DisplayIds} found. Aborting the process.
Entity Logic Validation	Payload has conflict with uniqueness on SourceSystemId or DisplayId with existing data	200	Duplicate General ledger accounts found '{DisplayId}' as it is having duplicate SourceSystemId or DisplayId

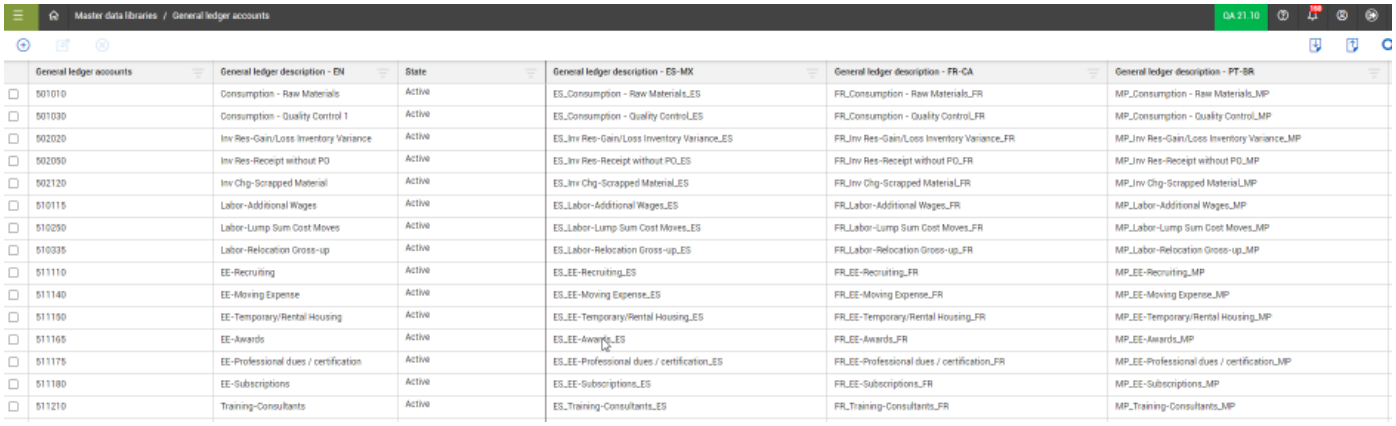
Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

Sample JSON

```
[{
   "DisplayId": "5211012",
   "Description": "Craft Administration",
   "SourceSystemId": "5211012",
   "SourceSystemName": "CLNTERPSOU",
   "IsAvailable": true,
   "Translations": [
      {
      "Description": "Craft Administration",
      "Language": "EN",
      }
   ]
   }
}]
```

Verification

Import General Ledger into InEight using the API AccountingLedger. The results can be validated in Main menu > Master data libraries > **General ledger accounts**.



General ledger accounts	General ledger description - EN	State	General ledger description - ES-MX	General ledger description - FR-CA	General ledger description - PT-BR
501010	Consumption - Raw Materials	Active	ES_Consumption - Raw Materials_ES	FR_Consumption - Raw Materials_FR	MP_Consumption - Raw Materials_MP
501030	Consumption - Quality Control 1	Active	ES_Consumption - Quality Control_ES	FR_Consumption - Quality Control_FR	MP_Consumption - Quality Control_MP
502020	Inv Res-Gain/Loss Inventory Variance	Active	ES_Inv Res-Gain/Loss Inventory Variance_ES	FR_Inv Res-Gain/Loss Inventory Variance_FR	MP_Inv Res-Gain/Loss Inventory Variance_MP
502050	Inv Res-Receipt without PO	Active	ES_Inv Res-Receipt without PO_ES	FR_Inv Res-Receipt without PO_FR	MP_Inv Res-Receipt without PO_MP
502120	Inv Chg-Scrapped Material	Active	ES_Inv Chg-Scrapped Material_ES	FR_Inv Chg-Scrapped Material_FR	MP_Inv Chg-Scrapped Material_MP
510115	Labor-Additional Wages	Active	ES_Labor-Additional Wages_ES	FR_Labor-Additional Wages_FR	MP_Labor-Additional Wages_MP
510250	Labor-Lump Sum Cost Moves	Active	ES_Labor-Lump Sum Cost Moves_ES	FR_Labor-Lump Sum Cost Moves_FR	MP_Labor-Lump Sum Cost Moves_MP
510335	Labor-Relocation Gross-up	Active	ES_Labor-Relocation Gross-up_ES	FR_Labor-Relocation Gross-up_FR	MP_Labor-Relocation Gross-up_MP
511110	EE-Recruiting	Active	ES_EE-Recruiting_ES	FR_EE-Recruiting_FR	MP_EE-Recruiting_MP
511140	EE-Moving Expense	Active	ES_EE-Moving Expense_ES	FR_EE-Moving Expense_FR	MP_EE-Moving Expense_MP
511150	EE-Temporary/Rental Housing	Active	ES_EE-Temporary/Rental Housing_ES	FR_EE-Temporary/Rental Housing_FR	MP_EE-Temporary/Rental Housing_MP
511165	EE-Awards	Active	ES_EE-Awards_ES	FR_EE-Awards_FR	MP_EE-Awards_MP
511175	EE-Professional dues / certification	Active	ES_EE-Professional dues / certification_ES	FR_EE-Professional dues / certification_FR	MP_EE-Professional dues / certification_MP
511180	EE-Subscriptions	Active	ES_EE-Subscriptions_ES	FR_EE-Subscriptions_FR	MP_EE-Subscriptions_MP
511210	Training-Consultants	Active	ES_Training-Consultants_ES	FR_Training-Consultants_FR	MP_Training-Consultants_MP

Incoterms

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Possibly hundreds of records to allow for all types of work supported by a customer.
APIM Name		Upsert IncoTerm
InEight Application	**Starting Version**	20.11
	Ending Version	

Fields

Depth	Name	Type	Precision	Parent	Req.
1	DisplayId	String	100		Yes
1	Description	String	255		Yes
1	SourceSystemId	String	50		No
1	SourceSystemName	String	50		No
1	IsAvailable	Boolean	NA		No
1	Translations	Array			
2	Description	String	255	Translations	No
2	Language	String	10	Translations	No

Field Descriptions

Name	Description	Example
Name	A code (usually 2 or 6 characters) that represents or describes the Incoterm. This is the Incoterms Code field in the UI.	CFR
Description	Customer-defined field that is typically used to describe or provide details about the Incoterm.	Costs and freight
SourceSystemId	Identification of the Incoterm used within the external system. In most cases, this matches the Name field. This field is used by InEight to match Incoterm records to other entities where an Incoterm is cross-referenced.	CFR
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	ERP
IsAvailable	Determines the status of the incoterm and whether it can be assigned. *True* equals available and *false* equals unavailable.	true

Name	Description	Example
Translations	Array Header for translation localized text.	
Description	Customer-defined field that is typically used to describe or provide details about the Incoterm as per the language code.	Costs and freight
Language	Language code that the default Incoterm is shown in the UI.	EN

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • Name • Description	400	The request is invalid. The [Field Name] Field is required.
API Validation	Payload exceeds String Length Size. [Field Name] options are: • Name • Description • Language • SourceSystemId • SourceSystemName	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length].
Entity Logic Validation	Payload has more than one record with same SourceSystemId	200	Duplicate Incoterm SourceSystemId: '{ SourceSystemId}' found. Skipping duplicate record.
Entity Logic Validation	Payload has more than one record with same Name	200	Duplicate Incoterm name: '{Name}' found. Skipping duplicate record.
Entity Logic Validation	Payload has conflict with uniqueness on SourceSystemId with existing data	200	SourceSystemId does not match with existing record : '{IncoTerm.Name}'. Skipping record.
Entity Logic Validation	Payload has conflict with uniqueness on Name with existing data	200	Incoterms code '{Name}' cannot be updated. Skipping record.

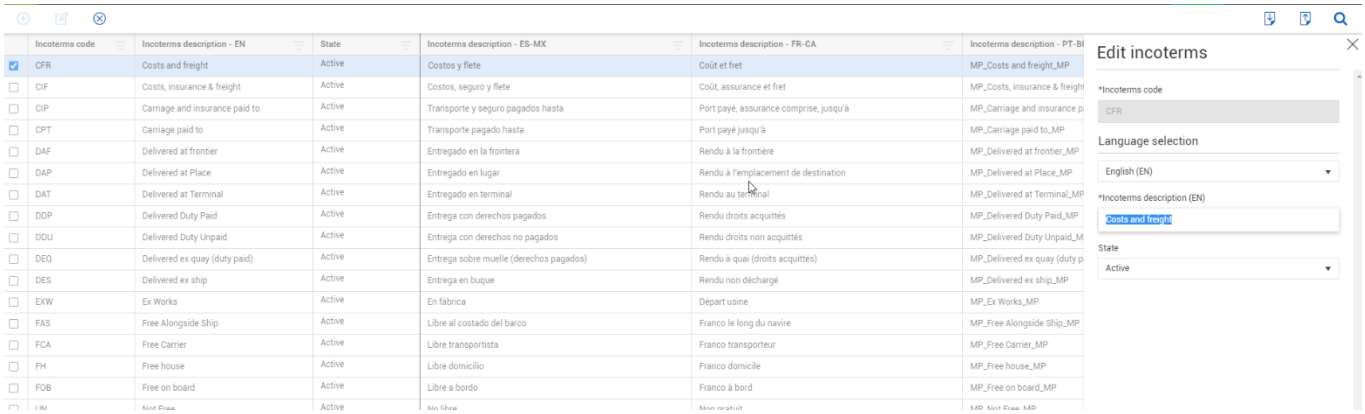
Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

Sample JSON

```
[{
   "DisplayId": "CFR",
   "Description": " Costs and freight ",
   "SourceSystemId": "CFR",
   "SourceSystemName": "ERP",
   "IsAvailable": true,
   "Translation": [
      {
      "Description": " Costs and freight",
      "Language": "EN",
      }
      ]
   }
}]
```

Verification

Import Incoterms into InEight using the API Upsert IncoTerm. The results can be validated in Main menu > Master data libraries > **Incoterms**.



Payment Terms

Direction		To InEight
Frequency		Determined by external system.
Trigger Methods		Determined by external system.
Average Payload Size		Possibly hundreds of records to allow for all types of work supported by a customer.
APIM Name		Upsert PaymentTerm
InEight Application	**Starting Version**	20.11
	Ending Version	

Fields

Depth	Name	Type	Precision[1]	Parent	Req.
1	Name	String	50		Yes
1	Description	String	500		Yes
1	SourceSystemId	String	50		No
1	SourceSystemName	String	50		No
1	IsAvailable	Boolean	NA		No
1	IsProgressEstimateNumberRequired	Boolean	NA		No
1	IsBillingPeriodRequired	Boolean	NA		No
1	DueDays	Number	8		No
1	Translations	Array			
2	Description	String	200	Translations	No
2	Language	String	10	Translations	No

Field Descriptions

Name	Description	Example
Name	A code (usually 2 or 6 characters) that represents or describes the payment term. This is the Payment Term Code field in the UI.	NT30
Description	Customer-defined field that is typically used to describe or provide details about the payment term.	Net due in 30 days
SourceSystemId	Identification of the payment term used in the external system. In most cases, this matches the Name field. This field is used by InEight to match Incoterm records to other entities where a payment term is cross-referenced.	NT30

Name	Description	Example
SourceSystemName	Unique name assigned to the system of record by the InEight cloud platform that matches the external system using the integration.	ERP
IsAvailable	Determines the status of the account/record and whether it can be assigned. *True* equals available and *false* equals unavailable.	true
IsProgressEstimateNumberRequired	Determines whether Progress Estimate number is mandatory in InEight Contract Payment Request module if the same Payment Term is used in the InEight Contract application.	true
IsBillingPeriodRequired	Determines whether Billing period is mandatory in InEight Contract Payment Request module if the same Payment Term is used in the InEight Contract application.	true
DueDays	When invoices are submitted in InEight Contract, the invoice due date will be determined based on the payment term due days.	30
PaymentTermText	Array header for localization text.	
Description	Customer-defined field that is typically used to describe or provide details about the payment term as per the language code.	Net due in 30 days
Language	Language code that the payment term is shown in the UI.	EN

Error Messages

The following error messages are generated by InEight for this integration. Errors in the table below are distinguished by the process that checks for the error.

- API Validation errors are basic record validations that will be returned to the API request message and cause the entire payload to fail.
- Entity Logic errors are performed internally within InEight and look for specific business rule or data integrity issues record-by-record. Failures with Entity Logic validations only cause the individual record to cease processing and are written to internal InEight logging.

API/Entity Logic	Condition	Code	Message
API Validation	Valid Payload	202	
API Validation	Required Fields are not Provided. [Field Name] options are: • Name • Description • DueDays • IsProgressEstimateNumberRequired • IsBillingPeriodRequired	400	The request is invalid. The [Field Name] Field is required.

API/Entity Logic	Condition	Code	Message
API Validation	Payload exceeds String Length Size. [Field Name] options are: • Name • Description • Language • DueDays • SourceSystemId • SourceSystemName	400	The request is invalid. The field [Field Name] must be a string with a maximum length of [String Length]. Message: The request is invalid.
Entity Logic Validation	Payload has more than one record with same SourceSystemId	200	Duplicate Payment term SourceSystem: {SourceSystemId} found. Aborting the process.
Entity Logic Validation	Payload has more than one record with same Name	200	Duplicate Payment term code: {PaymentTermNames} found. Aborting the process.
Entity Logic Validation	Payload has conflict with uniqueness on SourceSystemId or Name with existing data	200	Ignoring Payment term having Name '{ Name}' as it is having duplicate SourceSystemId or Name

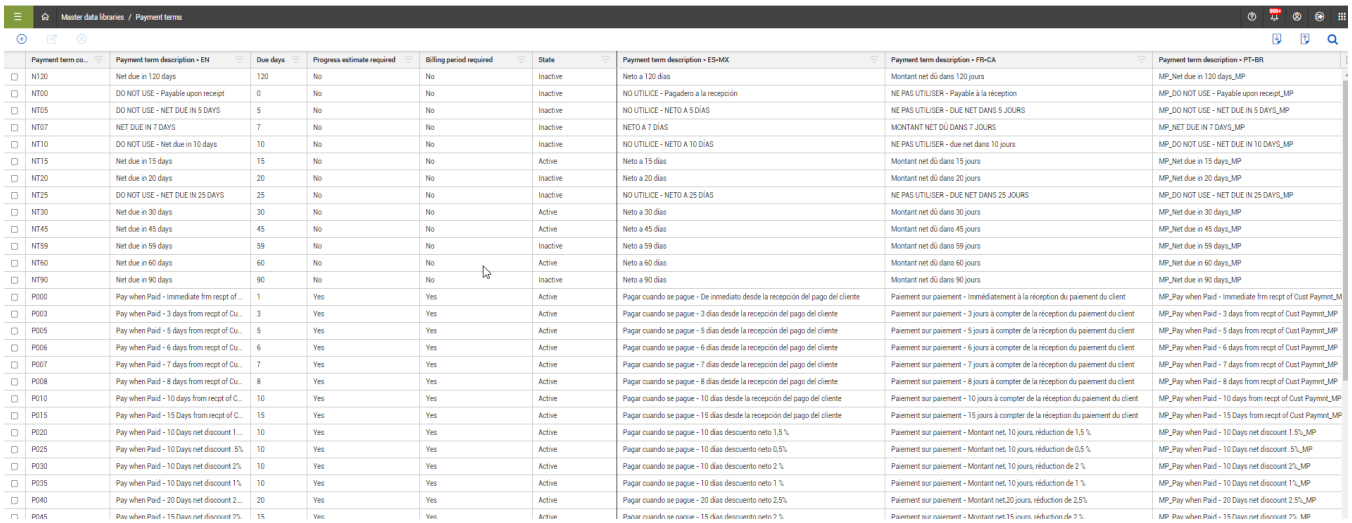
Entity Logic Validation	Valid Payload	200	{
			Logs: [ ]
			}

Sample JSON

```
[{
   "Name": "NT30",
   "Description": "Net due in 30 days",
   "SourceSystemId": "NT30",
   "SourceSystemName": "ERP",
   "IsAvailable": true,
   "IsProgressEstimateNumberRequired": true,
   "IsBillingPeriodRequired": true,
   "DueDays": 0,
   "Translations": [
      {
      "Description": "Net due in 30 days",
      "Language": "EN",
      }
   ]
   }
}]
```

Verification

Import Payment Terms into InEight using the API Upsert PaymentTerm. The results can be validated in Master data libraries > **Payment terms**.



Payment term co...	Payment term description - EN	Due days	Progress estimate required	Billing period required	State	Payment term description - ES-MX	Payment term description - FR-CA	Payment term description - PT-BR
N120	Net due in 120 days	120	No	No	Inactive	Neto a 120 días	Montant net dû dans 120 jours	MP_Net due in 120 days_MP
NT00	DO NOT USE - Payable upon receipt	0	No	No	Inactive	NO UTILICE - Pagadero a la recepción	NE PAS UTILISER - Payable à la réception	MP_DO NOT USE - Payable upon receipt_MP
NT05	DO NOT USE - NET DUE IN 5 DAYS	5	No	No	Inactive	NO UTILICE - NETO A 5 DÍAS	NE PAS UTILISER - DUE NET DANS 5 JOURS	MP_DO NOT USE - NET DUE IN 5 DAYS_MP
NT07	NET DUE IN 7 DAYS	7	No	No	Inactive	NETO A 7 DÍAS	MONTANT NET DÛ DANS 7 JOURS	MP_NET DUE IN 7 DAYS_MP
NT10	DO NOT USE - Net due in 10 days	10	No	No	Inactive	NO UTILICE - NETO A 10 DÍAS	NE PAS UTILISER - due net dans 10 jours	MP_DO NOT USE - NET DUE IN 10 DAYS_MP
NT15	Net due in 15 days	15	No	No	Active	Neto a 15 días	Montant net dû dans 15 jours	MP_Net due in 15 days_MP
NT20	Net due in 20 days	20	No	No	Inactive	Neto a 20 días	Montant net dû dans 20 jours	MP_Net due in 20 days_MP
NT25	DO NOT USE - NET DUE IN 25 DAYS	25	No	No	Inactive	NO UTILICE - NETO A 25 DÍAS	NE PAS UTILISER - DUE NET DANS 25 JOURS	MP_DO NOT USE - NET DUE IN 25 DAYS_MP
NT30	Net due in 30 days	30	No	No	Active	Neto a 30 días	Montant net dû dans 30 jours	MP_Net due in 30 days_MP
NT45	Net due in 45 days	45	No	No	Active	Neto a 45 días	Montant net dû dans 45 jours	MP_Net due in 45 days_MP
NT59	Net due in 59 days	59	No	No	Inactive	Neto a 59 días	Montant net dû dans 59 jours	MP_Net due in 59 days_MP
NT60	Net due in 60 days	60	No	No	Active	Neto a 60 días	Montant net dû dans 60 jours	MP_Net due in 60 days_MP
NT90	Net due in 90 days	90	No	No	Inactive	Neto a 90 días	Montant net dû dans 90 jours	MP_Net due in 90 days_MP
P000	Pay when Paid - Immediate frm recpt of ...	1	Yes	Yes	Active	Pagar cuando se pague - De inmediato desde la recepción del pago del cliente	Paiement sur paiement - Immédiatement à la réception du paiement du client	MP_Pay when Paid - Immediate frm recpt of Cust Paymnt_M
P003	Pay when Paid - 3 days from recpt of Cu...	3	Yes	Yes	Active	Pagar cuando se pague - 3 días desde la recepción del pago del cliente	Paiement sur paiement - 3 jours à compter de la réception du paiement du client	MP_Pay when Paid - 3 days from recpt of Cust Paymnt_MP
P005	Pay when Paid - 5 days from recpt of Cu...	5	Yes	Yes	Active	Pagar cuando se pague - 5 días desde la recepción del pago del cliente	Paiement sur paiement - 5 jours à compter de la réception du paiement du client	MP_Pay when Paid - 5 days from recpt of Cust Paymnt_MP
P006	Pay when Paid - 6 days from recpt of Cu...	6	Yes	Yes	Active	Pagar cuando se pague - 6 días desde la recepción del pago del cliente	Paiement sur paiement - 6 jours à compter de la réception du paiement du client	MP_Pay when Paid - 6 days from recpt of Cust Paymnt_MP
P007	Pay when Paid - 7 days from recpt of Cu...	7	Yes	Yes	Active	Pagar cuando se pague - 7 días desde la recepción del pago del cliente	Paiement sur paiement - 7 jours à compter de la réception du paiement du client	MP_Pay when Paid - 7 days from recpt of Cust Paymnt_MP
P008	Pay when Paid - 8 days from recpt of Cu...	8	Yes	Yes	Active	Pagar cuando se pague - 8 días desde la recepción del pago del cliente	Paiement sur paiement - 8 jours à compter de la réception du paiement du client	MP_Pay when Paid - 8 days from recpt of Cust Paymnt_MP
P010	Pay when Paid - 10 days from recpt of C...	10	Yes	Yes	Active	Pagar cuando se pague - 10 días desde la recepción del pago del cliente	Paiement sur paiement - 10 jours à compter de la réception du paiement du client	MP_Pay when Paid - 10 days from recpt of Cust Paymnt_MP
P015	Pay when Paid - 15 Days from recpt of C...	15	Yes	Yes	Active	Pagar cuando se pague - 15 días desde la recepción del pago del cliente	Paiement sur paiement - 15 jours à compter de la réception du paiement du client	MP_Pay when Paid - 15 Days from recpt of Cust Paymnt_MP
P020	Pay when Paid - 10 Days net discount 1...	10	Yes	Yes	Active	Pagar cuando se pague - 10 días descuento neto 1,5 %	Paiement sur paiement - Montant net, 10 jours, réduction de 1,5 %	MP_Pay when Paid - 10 Days net discount 1.5%_MP
P025	Pay when Paid - 10 Days net discount .5%	10	Yes	Yes	Active	Pagar cuando se pague - 10 días descuento neto 0,5%	Paiement sur paiement - Montant net, 10 jours, réduction de 0,5 %	MP_Pay when Paid - 10 Days net discount .5%_MP
P030	Pay when Paid - 10 Days net discount 2%	10	Yes	Yes	Active	Pagar cuando se pague - 10 días descuento neto 2 %	Paiement sur paiement - Montant net, 10 jours, réduction de 2 %	MP_Pay when Paid - 10 Days net discount 2%_MP
P035	Pay when Paid - 10 Days net discount 1%	10	Yes	Yes	Active	Pagar cuando se pague - 10 días descuento neto 1 %	Paiement sur paiement - Montant net, 10 jours, réduction de 1 %	MP_Pay when Paid - 10 Days net discount 1%_MP
P040	Pay when Paid - 20 Days net discount 2...	20	Yes	Yes	Active	Pagar cuando se pague - 20 días descuento neto 2,5%	Paiement sur paiement - Montant net,20 jours, réduction de 2,5%	MP_Pay when Paid - 20 Days net discount 2.5%_MP
P045	Pay when Paid - 15 Days net discount 2%	15	Yes	Yes	Active	Pagar cuando se pague - 15 días descuento neto 2 %	Paiement sur paiement - Montant net,15 jours, réduction de 2 %	MP_Pay when Paid - 15 Days net discount 2%_MP