

# Performed Work and Payroll Integration Specification



InEight  
PLAN



InEight  
PROGRESS



## 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
0.01	16-AUG-2018	Added information for Daily Plan Status and Time Card Confirmation
1.00	10-SEP-2018	Published release
		<p>Added CompleteMessage filter for DailyPlan_Get</p> <p>Added additional information for the description of DailyPlan_Get to note that it returns daily plans with a status of Approved.</p> <p>Changed title of Work Orders to <a href="#">Daily Plan Work Orders</a> (Post) to clarify the relationship to the integration in APIM and the Integration Catalog.</p> <p>Updated <a href="#">Daily Plan Status</a> to include information on fields not originally documented. New sample and verification information added.</p>
		Updated Time Card to reflect changes since original documentation. Fields, Descriptions, and Sample JSON updated.
2.00	25-OCT-2018	Published revision.
		Added DailyPlanMaintenance array to <a href="#">Daily Plans</a> . Restructured Daily Plans to describe push and pull options. Updated field descriptions in <a href="#">Daily Plan Work Orders</a> . Added Data Availability section and process flow to <a href="#">Time Card</a> .
		Moved content into latest template. Added <a href="#">Quantity Claiming Detail</a> . Added reason codes to <a href="#">Daily Plans</a> .
		Added information for data sourcing and Send selected option in <a href="#">Time Card</a> . Added initial Component Details API information.
		Updates to <a href="#">Daily Plans</a> to improve clarity and for references to data associations.
3.00	18-JUN-2019	Published revision.
		<p>Updated information about <a href="#">resynchronizing Daily Plans</a></p> <p>New fields for Shift information added to <a href="#">Daily Plans integration</a></p>
4.00	02-OCT-2019	Published revision.
5.0	03-MAR-2020	<p>Updated <a href="#">Daily Plans</a>, added <a href="#">ShiftDetails</a> section.</p> <p>Published revision.</p>
6.0	15-MAY-2020	<p>Updated <a href="#">Daily Plan</a>, added DailyPlanNoteMaintenance array and associated fields.</p> <p>Published revision (Release 20.5).</p>
7.0	07-AUG-2020	<p>Added details to <a href="#">Time Card</a> overview. Updated Note and diagram in <a href="#">Data Availability</a> for time card processing. Added <a href="#">ShiftDetails</a> array and associated fields to Time Card.</p> <p>Added Note to <a href="#">Daily Plan</a> overview and updated flow diagram. Updated <a href="#">resync criteria</a>. Updated <a href="#">Daily Plan Pull</a> overview.</p> <p>Published revision (Release 20.7).</p>

8.0	05-MAR-2021	Updated <a href="#">Time Card</a> field descriptions for Craft and Trade logic changes. Published revision (Release 21.1)
9.0	13-APR-2021	Added BatchSize field in <a href="#">Time Card</a> , referenced batch size parameter in <a href="#">Data Availability</a> , and added BatchSize in <a href="#">Supported Filters</a> . Added array and associated fields for <a href="#">DailyPlanSignin</a> and <a href="#">DailyPlanBreaks</a> in Daily Plan and updated <a href="#">Appendix A</a> . Published revision (Release 21.3)
10.0	17-SEP-2021	Added TimeCardType field in <a href="#">Time Card</a> . Published revision (Release 21.7)
11.0	25-OCT-2021	Add Employee Billing Class field to <a href="#">Time Card</a> . Add Billing Class field to <a href="#">Daily Plans</a> . Published revision (Release 21.10)
12.0	04-MAR-2022	Changed CostItemId field type from string to number in <a href="#">Daily Plan Note</a> . Published revision (Release 22.2)
13.0	29-JUL-2022	Added flow diagram in <a href="#">Daily Plan versus Time Card</a> to show differences between integrations. Added DailyPlanCostItemVendors array and associated fields to <a href="#">Daily Plan Cost Item</a> . Added DailyPlanNoteVendors array and associated fields to <a href="#">Daily Plan Note</a> . Published revision (Release 22.6)
14.0	23-AUG-2022	Added EmployeeUDF1 field to <a href="#">Time Card</a> . Added ApproverPosition field, and DailyPlanBreakDetails array and fields to <a href="#">Daily Plans</a> . Published revision (Release 22.8)
15.0	09-JAN-2023	Updated the decimal precision of quantity related fields in <a href="#">Daily Plans</a> , <a href="#">Quantity Claiming Detail</a> , and <a href="#">Component Details</a> from 16,5 to 28,15 . Published revision (Release 22.12)
16.0	16-MAR-2023	Added ProjectId, WBSPHaseCode, and BillingCode fields to <a href="#">Time Card</a> . Added BillingCode field to the ReasonCode array in <a href="#">Daily Plan Cost Item</a> . Published revision (Release 23.2)
17.0	24-AUG-2023	Added <a href="#">Components</a> integrations. Published revision (Release 23.6)



# Contents

- Overview .....1**
- Integrations in this Document .....1
- Relationships and Dependencies .....2
- Daily Plan versus Time Card..... 2
- Prerequisites .....2
- Related Integrations .....3
- Time Card .....4**
- Data Availability .....4
- Supported Filters ..... 6
- Fields .....6
- Field Descriptions.....7
- Error Messages ..... 11
- Sample JSON ..... 11
- Time Card Confirmation .....13**
- Fields ..... 13
- Field Descriptions..... 13
- Error Messages ..... 14
- Sample JSON ..... 14
- Verification..... 15
- Daily Plans .....16**
- Resynchronizing Daily Plans..... 17
- Daily Plan Push..... 18
- Daily Plan Pull..... 19
- Supported Filters ..... 19
- Fields ..... 20
- Daily Plan Detail..... 20
- Daily Plan Cost Item..... 22
- Daily Plan Note ..... 24
- Daily Plan Signoff..... 25
- Daily Plan Signin..... 25

Daily Plan Breaks ..... 26

Daily Plan Break Details ..... 26

Employee Level Breaks ..... 26

Executors ..... 26

Approvers ..... 27

Daily Plan Employee Payroll Indicator ..... 27

Daily Plan Client Signoff ..... 27

Daily Plan Maintenance ..... 27

Shift Details ..... 28

Field Descriptions ..... 28

Sample JSON ..... 39

    Example 1: Daily Plan with Reason Code hours ..... 39

    Example 2: Daily Plan with components and hours worked in Overtime, Double time, Standard time ..... 45

**Daily Plan Status ..... 51**

    Fields ..... 51

    Field Descriptions ..... 51

    Error Messages ..... 52

    Sample JSON ..... 52

    Verification ..... 53

**Employee Work Schedule ..... 54**

    Fields ..... 54

    Field Descriptions ..... 55

    Error Messages ..... 55

    Sample JSON ..... 56

    Verification ..... 56

**Daily Plan Work Orders (Post) ..... 57**

    Fields ..... 57

    Field Descriptions ..... 58

    Error Messages ..... 59

    Sample JSON ..... 59

    Verification ..... 60

**Quantity Claiming Detail (Get).....61**

    Fields ..... 62

    Field Descriptions..... 63

    Sample JSON ..... 64

**Component Details .....65**

    Supported Filters..... 65

    Fields ..... 65

    Field Descriptions..... 68

    Sample JSON ..... 70

**Components .....73**

    Fields ..... 73

    Field Descriptions..... 76

    Sample JSON ..... 79

**Appendix A: Full Daily Plan Schema .....82**





## Overview

This group of APIs provide information from InEight Plan about planned or completed daily work and information that might be required to fulfill payroll obligations.

### Integrations in this Document

Integration	Description
Time Card	<p>The Time Card integration provides payroll or pre-payroll processing systems with a detailed breakdown of hours for employees along with the type of work performed and custom-defined that are applied for blocks of hours or for the entire day. This integration is available if Time Center has been enabled in the customer environment.</p> <p>Time Center is an application in InEight Plan that generates time cards of individual employees or equipment for each combination that is entered on a daily plan. The application allows an administrator to review the hours before sending it to an ERP for further processing of actual hours for other business processes such as payroll. All time cards, when required, can be sent through the Time Card integration to an ERP or other system for further business process workflows.</p>
Time Card Confirmation	<p>The Time Card Confirmation integration allows external systems to report on the success or failure to process time cards obtained from Time Center and update a status to support a business workflow.</p>
Daily Plan	<p>The Daily Plan integration allows customers to provide time keeping information for employees and equipment as recorded in InEight Plan. This integration includes summary information about the daily plan, from employee hours to cost items, equipment hours to cost items, payroll indicators, and notes.</p> <p>The Daily Plan integration can also be used to track and provide plan status in systems outside of InEight cloud platform. When a daily plan is used for this purpose, an optional response message can be used to trigger a status change in the daily plan to indicate the change has been successfully received by the external system.</p>
Daily Plan Status	<p>If approved daily plans are sent to an external system, the external system can optionally process the daily plan data and return a message informing the InEight cloud platform that the status should be updated to Final Processing Complete.</p>
Employee Work Schedule	<p>This integration allows customers to provide a simple listing of the scheduled days when employees should be at work and for the number of hours they are scheduled. This information is used for lookup on validations that employee data entered in a daily plan (employee, date, hours) falls in expected parameters of their planned work schedule.</p>
Daily Plan Work Orders	<p>This integration allows a master list of work orders generated by an external system to be maintained in InEight Plan. Work orders can be selected in a daily plan when recording employee hours to show what the employee was specifically working on and relate that effort back to planned maintenance activities and estimated maintenance costs.</p>
Quantity Claiming Detail	<p>This integration allows customers to retrieve all individual claiming transactions that have been approved from daily plans or quantity tracking in InEight Plan and InEight Progress.</p>
Component Details	<p>The Component Details integration allows customers to retrieve a complete list of all Components and their details including custom fields (characteristics) from InEight Plan.</p>
Components	<p>These integrations allow customers to create, update, and request all Plan components and custom characteristics between their internal systems and the InEight cloud platform.</p>

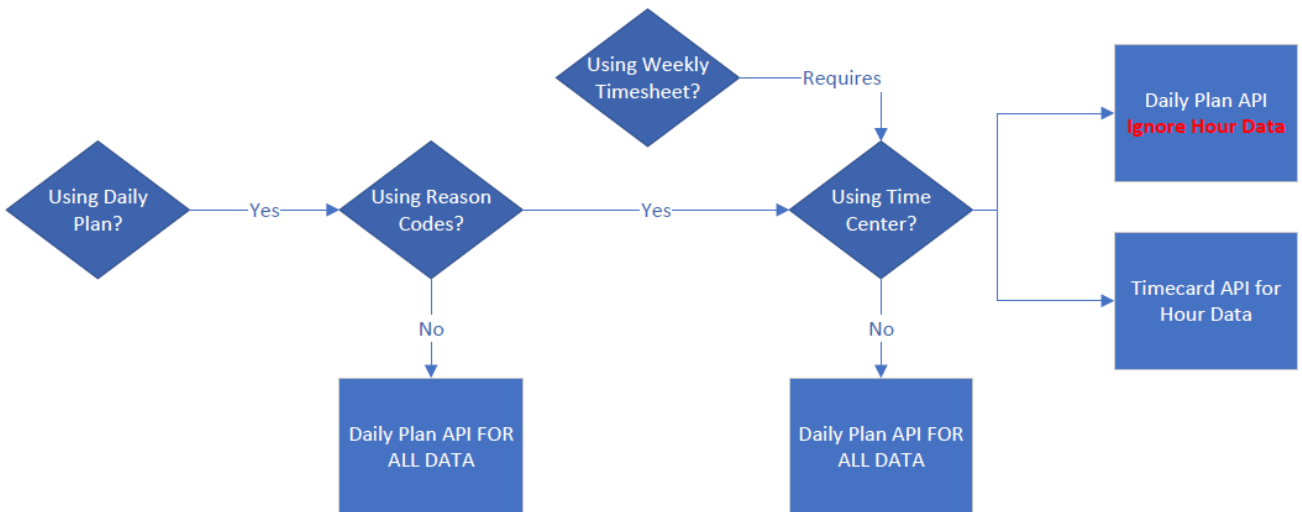
## Relationships and Dependencies

### Daily Plan versus Time Card

The Daily Plan integration differs from the Time Card integration in that the Time Card integration primarily functions to provide the hours of individual employees or equipment included in the daily plan. This information is broken down into separate time card records by project and by type of hours.

The following diagram can be used to help determine and decide which integration, either Daily Plan or Time Center, is applicable for the customer’s needs.

Daily Plan API – What does it contain?	As of the time of Daily Approval: Employee/Equipment/Maintenance Hours Allowances Employee Signoffs Client Signoff Notes Quantities	Timecard API – What does it contain?	Timecards for approved Weekly Timesheets and Daily Plans that have been edited:  Employee/Equipment/Maintenance Hours Allowances
Daily Plan API – When is data generated?	When Foreman or Superintendent approves the Daily Plan	Timecard API – When is data generated?	1.) On Client approval (if configured)  2.) When Project Accountant selects a timecard version to send to payroll
Which API should I use?			



### Prerequisites

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

Prerequisite	Description	Required by
EmployeeId	<p>EmployeeId in this integration must match a valid SourceSystemId for an employee record in the InEight cloud platform.</p> <p>Employees are provided to the InEight cloud platform either via integration (refer to <i>InEight Master Data for Employees Integration Specification</i> and <i>InEight Project Setup and Maintenance Integration Specification</i>) or manually entered through the UI.</p>	Employee Work Schedule
EquipmentSourceSystemId	<p>A valid SourceSystemId for an equipment record created in InEight cloud platform.</p> <p>Equipment can be provided to InEight cloud platform either via integration (refer to <i>InEight Master Data for Equipment Integration Specification</i> and <i>InEight Project Setup and Maintenance Integration Specification</i>) or manually entered through the UI.</p>	Daily Plan Work Orders
WorkOrderTypeDisplay	Data provided in this field will be verified against a list of values in the InEight cloud platform. See the field description in DailyPlanWorkOrder for more details of allowed values.	Daily Plan Work Orders
ProjectDisplay	Valid DisplayId for a project in InEight cloud platform.	Daily Plan Work Orders

## Related Integrations

Integration Name	Description	Document Name
Employees	Creates and maintains the master list of all employees for an account.	Master Data for Employees
Project Employees	Provides the list of all employees assigned to a project that can be selected for a daily plan.	Project Initiation and Maintenance
Equipment	Creates and maintains the master list of all equipment for an account.	Master Data for Equipment
Project Equipment	Provides the list of all equipment assigned to a project that can be selected for a daily plan.	Project Initiation and Maintenance

## Time Card

The Time Card integration provides payroll or pre-payroll processing systems with a detailed breakdown of hours for employees, along with the type of work performed and any custom-defined names that are applied to blocks of hours or for the entire day.

**NOTE:** The Time Card integration is only available when using Time Center, and it must be enabled as a separate option in the customer environment.

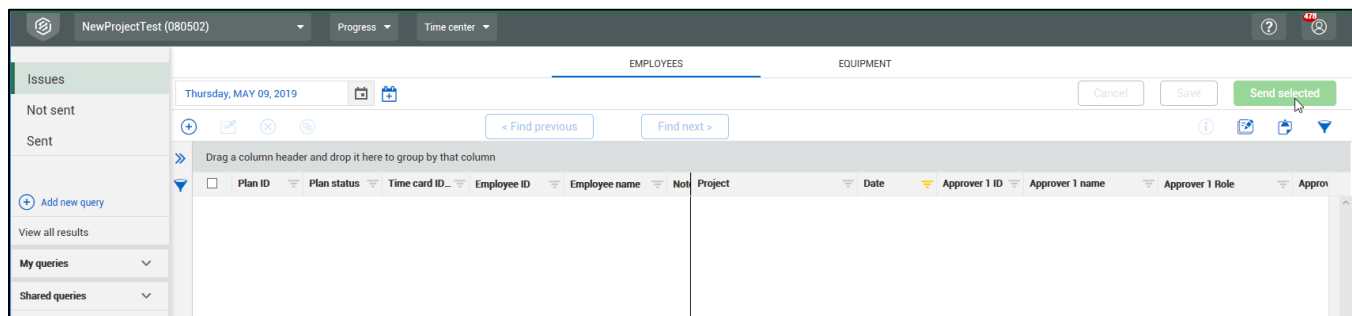
Time Center is a module within the InEight Progress application that generates timecards of individual employees or equipment for each combination that is entered in a daily plan. The application allows an administrator to review the hours before sending it to an ERP for further processing of actual hours for other business processes such as payroll. All timecards, when required, can be sent through the Time Card integration to an ERP or other system for further business process workflows.

When using this integration to obtain payroll data for reported hours and additional payroll processing codes, it is required to use the Time Card Confirmation integration to report the success or failure of receipt of data.

<b>Direction</b>		From the InEight cloud platform.
<b>Frequency</b>		Manually triggered as needed or could be a scheduled pull from external systems.
<b>Trigger Methods</b>		Pull initiated from external system.
<b>Average Payload Size</b>		Up to 200 time cards in a batch.
<b>APIM Name</b>		TimeCard_Get
<b>InEight Application</b>	<b>Starting Version</b>	18.2
	<b>Ending Version</b>	

## Data Availability

The data for time card records comes in to Time Center from the creation and approval of daily plans and weekly time sheets in InEight Progress. When a user in Time Center selects records, and then clicks **Send selected**, those records are placed into an external queue, which makes them available for external systems to retrieve.



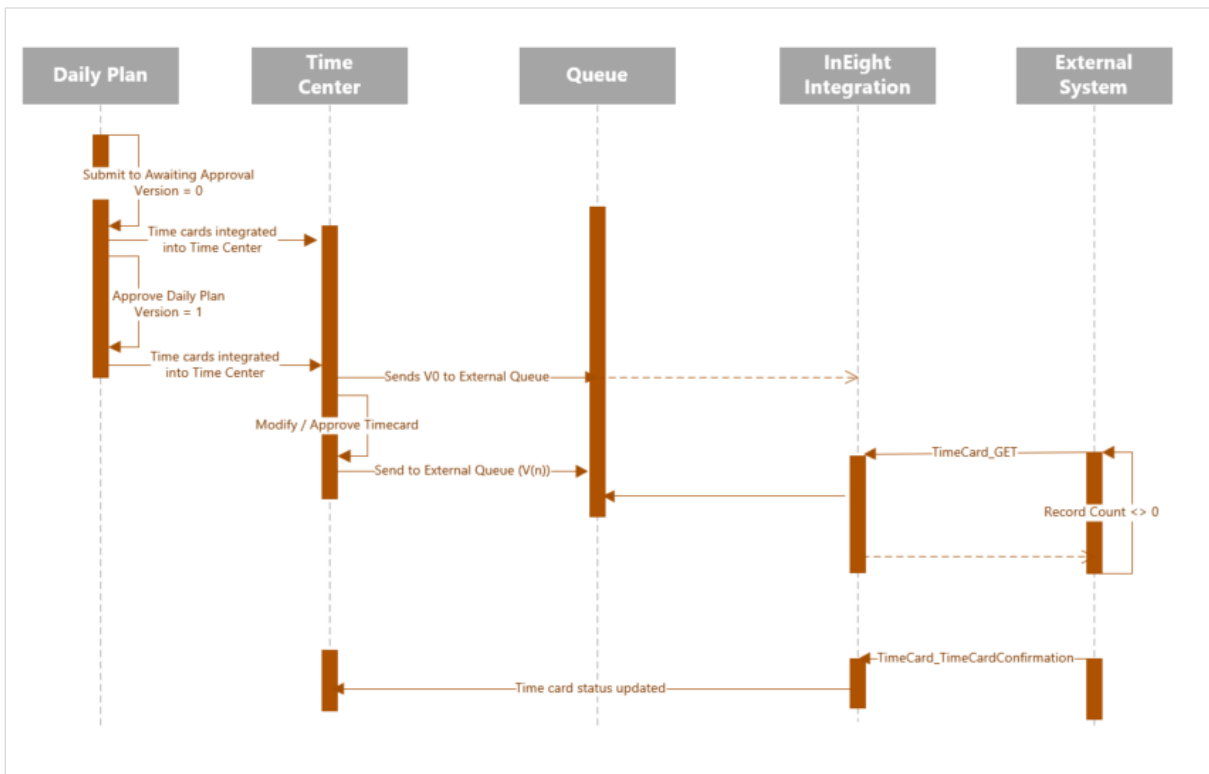
However, several status and timing events determine eligibility and timing of when a time card is placed in the queue and what specific data is available.

- When a daily plan status is changed from Execution to Awaiting Approval by a foreman in the InEight Plan or InEight Progress applications, time card records from the daily plan are created with a version of 0 (zero) but are not yet placed in the external queue in case the daily plan is sent back.
- When the daily plan status is changed from Waiting Approval to Approved, the version 0 time cards related to the daily plan are placed in the external queue with the data recorded at the time of the Execution to Awaiting Approval status change. New time card records are also created as Version 1, containing data recorded from the Approval status change. Version 1 time cards are not placed in the external queue but made available to the Time Center application for further review.
- Within Time Center, time card records are reviewed and possibly changed. Each update to a time card increments the version number. A user in Time Center must be specific to indicate the time cards to send, which then places them into the external queue with the applicable version number.

**NOTE:** Records that are placed in the external queue are removed after they are downloaded successfully.

When retrieving time cards from the queue, the batch size parameter determines the number of records returned per request, which can be set from 25 to 2000 records per request (default is 200 records). If more records are in the queue than what is set in the batch size, the external system should make additional requests to retrieve additional timecards. The external system should continue making requests, until the entire queue is cleared, and no records are returned from the InEight (cloud platform) integration.

When the external system confirms the receipt of the time card, the status will be updated on the time card, and the same status is then reflected in the Time Center application.



## Supported Filters

Data provided by the InEight cloud platform to external systems (outbound), can support selective fetching of data by applying filters in the API request.

Filter Name	Data Type	Description
BatchSize	Integer	Optional: Format: Int32 Specifies the number of records to return for the request in increments of 25. Minimum allowed value for this filter is 25. Maximum allowed value for this filter is 2000.

## Fields

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	PlanId	Number	19,0	
1	PlanDate <sup>3</sup>	String	34	
1	PlanTitle	String	100	
1	TimeCardId <sup>2</sup>	String	21	
1	TimeCardType	String	21	
1	CreatedDate <sup>3</sup>	String	34	
1	ModifiedDate <sup>3</sup>	String	34	
1	Executor	String	50	
1	ShiftDetails	Array	NA	
2	Shift	String	100	ShiftDetails
2	ShiftStartDateTime	String	25	ShiftDetails
2	ShiftEndDateTime	String	25	ShiftDetails
2	EmployeeShiftDetails	Array	NA	ShiftDetails
3	ShiftStartDateTime	String	25	EmployeeShiftDetails
3	ShiftEndDateTime	String	25	EmployeeShiftDetails
1	Trade	String	50	
1	Craft	String	50	
1	EmployeeBillingClass	String	50	
1	EmployeeId	String	50	
1	EquipmentId	String	50	
1	EmployeeReasonCode	String	50	

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	EquipmentReasonCode	String	50	
1	EmployeeHours	Decimal	16,5	
1	EquipmentHours	Decimal	16,5	
1	Version <sup>2</sup>	Decimal	10,0	
1	Segment1	String	50	
1	Segment2	String	50	
1	Segment3	String	50	
1	Segment4	String	50	
1	WorkOrder	String	50	
1	MaintenanceEquipmentId	String	50	
1	OverriddenTrade	String	50	
1	OverriddenCraft	String	50	
1	Premiums	Array List	NA	
1	TimeCardCharacteristic	Array	NA	
2	Name	String	50	TimeCardCharacteristic
2	Value	String	50	TimeCardCharacteristic
1	AdditionalPayrollInstructionCodeOverridden	Boolean	NA	
1	IsActive	Boolean	NA	
1.	EmployeeUDF1	String	250	
1	ProjectId	String	200	
1	WBSPhaseCode	String	50	
1	BillingCode	String	10	

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
PlanId	Unique identifier of the daily plan where the hours were reported for the employee.	1234
PlanDate	Date on which the daily plan hours were performed.	2018-05-31T00:00:00+00:00

Name	Description	Example
PlanTitle	Name given to the daily plan where the hours were entered.	Excavation - Mike's crew
TimeCardId	Unique identifier of the time card record.	RD1234, PI3784
TimeCardType	Indicates the type of timecard that is being sent. Can be one of the following: Labor, Equipment, Labor-Equipment, Administrative, or Maintenance.	Labor
CreatedDate	Date when this specific time card version, was originally created.	2018-02-20T14:47:05.9833049+00:00
ModifiedDate	Date when this specific time card version was last modified.	2018-02-20T14:58:47.4779635+00:00
Executor	Unique identifier of the employee representing the role of Executor on the daily plan	foreman@company.com
ShiftDetails	Array of the shift details represented by the daily plan.	
Shift	Name or abbreviated code of the shift represented by the daily plan. Shift names are custom defined in InEight Plan settings.	Day
ShiftStartDateTime	Default date and time the crew began work for a specific daily plan.	2018-06-29T00:00:00Z
ShiftEndDateTime	Default date and time the crew completed work for a specific daily plan.	2018-06-30T00:00:00Z
EmployeeShiftDetails	Array of employees that worked the shift on the daily plan.	
ShiftStartDateTime	Date and time a specific employee began work for a specific daily plan.	2018-06-29T00:00:00Z
ShiftEndDateTime	Date and time a specific employee completed work for a specific daily plan.	2018-06-30T00:00:00Z
Trade	Unique identifier from the trade system of record for the employee/time card. Can be one of the following: <ul style="list-style-type: none"> <li>• If available, use trade associated to Project Employee Craft (rate code).</li> <li>• If not available, use Employee Trade.</li> </ul>	CARP
Craft	Unique identifier from the craft system of record for the employee/ time card. Can be one of the following: <ul style="list-style-type: none"> <li>• If available, use Project Employee Craft (rate code).</li> <li>• If not available, use Employee Craft.</li> </ul>	APP1
EmployeeId	Unique identifier from the HR system of record of the employee. This must match the employee's SourceSystemId.	301847



Name	Description	Example
EmployeeBillingClass	Description of the type of work performed by the employee that is used to bill the client.	Operator 1
EquipmentId	Equipment that was operated while performing labor operations for reported hours. This value is based on the SourceSystemId received from an external system.	478933
EmployeeReasonCode	One of the following: <ul style="list-style-type: none"> <li>• Reason code for labor hours,</li> <li>• Reason code for allowances if the time card is for allowance only.</li> </ul> Each allowance given to an employee for the day will be on a separate time card with zero hours. Each reason code/hours combination for an employee are on a separate time card. <b>NOTE:</b> The exception is when there are hours worked and equipment operated (linked). In this case, the employee reason code is in this field and the equipment reason code is in the equipment reason code field on the same time card.	AWK, PTO
EquipmentReasonCode	If the hours entered in a daily plan are for operated equipment and a reason code was selected for the block of hours, the reason code will be reported here.	OPT, DWN
EmployeeHours	This field is used to for employee activities hours that are entered on the time card. When the time card represents work against a work order, the hours spent on the work order is shown here.	7.50, 4.25
EquipmentHours	Hours reported for equipment being operated during labor operations.	7.50
Version	Current version of the Time Card. Versioning starts at 0 and is incremented each time it is resent to the receiving system.	1
Segment1	First segment of the cost item associated with the hours.	104354
Segment2	Second segment of the cost item associated with the hours.	104354A
Segment3	Third segment of the cost item associated with the hours.	4093
Segment4	Fourth segment of the cost item associated with the hours.	
WorkOrder	Unique identifier for a work order if one was selected for the hours. This value is based on the SourceSystemId received from an external system.	EQ12387

Name	Description	Example
MaintenanceEquipmentId	Unique identifier for the equipment that was either specified by a work order, or the selected equipment for maintenance hours. A value for this field is only present if a work order was selected for the hours. This value is based on the SourceSystemId received from an external system.	834698
OverrideTrade	If the trade has been overridden on the time card from the default trade assigned to the employee, this field will show the overridden TradeId value.	ELECT
OverrideCraft	If the craft has been overridden on the time card from the default craft assigned to the employee, this field will show the overridden CraftId value.	APP1
Premiums	Array for the list of all premium code payroll indicators belonging to the premiums category that were assigned to this specific time card version.	D/N
TimeCardCharacteristics	Array header for the custom-defined fields used in the daily plan and associated with the hours in the time card. In this array, any characteristic name/value pair can be repeated if there are hours in the time card for multiple resource types (e.g. Labor and Equipment).	NA
Name	Name of the time card characteristic for which the value is being set. This field can be repeated as needed in the TimeCardCharacteristic list. Available options for name are EmployeeCostType and EquipmentCostType	EquipmentCostType
Value	Value of the time card characteristic name. This field can be repeated for each characteristic name provided.	1940
AdditionalPayrollInstructionCodeOverridden	If the Allow Additional Payroll Instruction Code for the employee has been modified (changed or removed) for the time card for any reason, this field will show as <i>true</i> .	false
IsActive	Sending a value of false in this field causes the Time Card record to be soft deleted from the InEight cloud platform. If a value is not provided, the default value true is used. <b>NOTE:</b> Records can be deleted even after they are integrated with payroll. In this case, the system automatically sends the deleted time card to payroll.	true
EmployeeUDF1	Value of project employee user defined field.	50.00
ProjectId	Project number that is shown in the UI.	105164

Name	Description	Example
WBSPHaseCode	WBS phase code associated with hours for the employee.	1234
BillingCode	Billing code that is used for mapping to the InEight Billings module.	ST01

## Error Messages

There are no error messages provided for this outbound integration.

## Sample JSON

```
[
  {
    "PlanId": 1,
    "PlanDate": "2018-09-27T00:00:00+00:00",
    "PlanTitle": "DCB Plan 4 July 2018",
    "TimecardId": "RD5",
    "TimeCardType": "Labor",
    "CreatedDate": "2018-09-27T16:55:58.0454023+00:00",
    "ModifiedDate": "2018-09-27T17:05:14.6130476+00:00",
    "Executor": "foreman@company.com",
    "ShiftDetails": [
      {
        "Shift": "Day",
        "ShiftStartDateTime": "2018-06-29T00:00:00+00:00",
        "ShiftEndDateTime": "2018-06-30T00:00:00+00:00",
        "EmployeeShiftDetails": [
          {
            "ShiftStartDateTime": "2018-06-29T00:00:00+00:00",
            "ShiftEndDateTime": "2018-06-30T00:00:00+00:00"
          }
        ]
      }
    ]
  },
  "Trade": "CARP",
  "Craft": "APP1",
  "EmployeeBillingClass": "Operator 1",
  "EmployeeId": "301847",
  "EquipmentId": "6546477",
  "EmployeeReasonCode": "AWK",
  "EquipmentReasonCode": "OPT",
  "EmployeeHours": 7.50,
  "EquipmentHours": 7.50,
  "Version": 1,
  "Segment1": "104354",
  "Segment2": "104354A",
  "Segment3": "4093",
  "Segment4": "1234",
  "WorkOrder": "EQ12387",
  "MaintenanceEquipmentId": "834698",
  "OverrideTrade": "ELECT",

```

```
"OverrideCraft": "APP1",
"Premiums": [
  "D/N"
],
"TimeCardCharacteristics": [
  {
    "Name": "EquipmentCostType",
    "Value": "1022"
  }
],
"AdditionalPayrollInstructionCodeOverridden": false
"IsActive": "true"
}
"EmployeeUDF1": "50.00"
"ProjectId": "105164"
"WBSPhaseCode": "1234"
"BillingCode": "ST01"
]
```

## Time Card Confirmation

The Time Card Confirmation Import integration allows external systems to report on the success or failure to process time cards obtained from Time Center and update a status to support a business workflow.

<b>Direction</b>	To the InEight cloud platform	
<b>Frequency</b>	Determined by external system	
<b>Trigger Methods</b>	Determined by external system	
<b>Average Payload Size</b>	Receipt of all time cards, resulting in a large set of records, or sent as time cards are processed by an external system, resulting in one or a few records per message.	
<b>APIM Name</b>	TimeCardConfirmation_Import	
<b>InEight Application</b>	<b>Starting Version</b>	18.2
	<b>Ending Version</b>	

## Fields

Depth	Name	Type	Precision	Parent	Req.
1	TimeCardId	String	50		Yes
1	Version	String	50		Yes
1	Status	String	50		Yes
1	Errors	Array	List		No

## Field Descriptions

Name	Description	Example
TimeCardId	Unique identifier of the time card. This is version specific.	RD13444
Version	Version of the time card. Each time a change is saved for a time card, the version number is incremented by one.	1
Status	Complete or Failed	Complete
Errors	Array of the list of error messages that should appear if the time card cannot be processed by the external system.	Duplicate Time Card record.

## Error Messages

The following error messages are generated by the InEight cloud platform and products 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 in the InEight cloud platform and products to 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 logging.

API/Entity Logic	Condition	Code	Message
API Validation	All received records have been validated.	200	
API Validation	Time Card Id is not present in a record. Each failed record will have a separate message in the response body.	200	An Empty/Blank time card Id found and that confirmation is ignored.
API Validation	Version is not present in a record. Each failed record will have a separate message in the response body.	200	An Empty/Blank Version found and the confirmation for {TimeCardId} is ignored
API Validation	Version contains an invalid value in a record. Each failed record will have a separate message in the response body.	200	Invalid value found in the version and the confirmation for {TimeCardId} is ignored.

## Sample JSON

```
[
  {
    "TimeCardId": "PI32",
    "Version": 1,
    "Status": "Complete",
    "Errors": []
  }
]
[
  {
    "TimeCardId": "PI32",
    "Version": 1,
    "Status": "Failed",
    "Errors": [
      "Duplicate Time Record Found",
      "Employee Not Scheduled for Day"
    ]
  }
]
```

## Verification

1. Open Time Center
2. Search for a time card used for testing.
3. Trigger external system to GET Time Cards.
4. External system returns Confirmation.
5. Refresh Time Center and search for time card.
6. Verify Time Card status matches test data returned in Confirmation.
  - a. Complete will display as Sent – Current.
  - b. Failed will display as Failed.

**NOTE:**

If a time card being tested is edited and saved, the changes become a new time card version and must be Sent and Confirmed separately from any other version of the same time card.

## Daily Plans

Integration of daily plans to an external system can be configured in one of two methods. These methods are not compatible with each other, therefore only one method can be used per customer.

- The push method is configured in InEight cloud platform as a web service and does not have a corresponding API available in APIM. This method automatically sends a daily plan to a configured end point whenever it is set to a status of approved in the InEight Plan or InEight Progress UI.
- The pull method places daily plans into an integration queue when they are set to a status of Approved in the UI. External systems can then use the DailyPlan\_Get API in APIM to retrieve daily plans from the queue on demand.

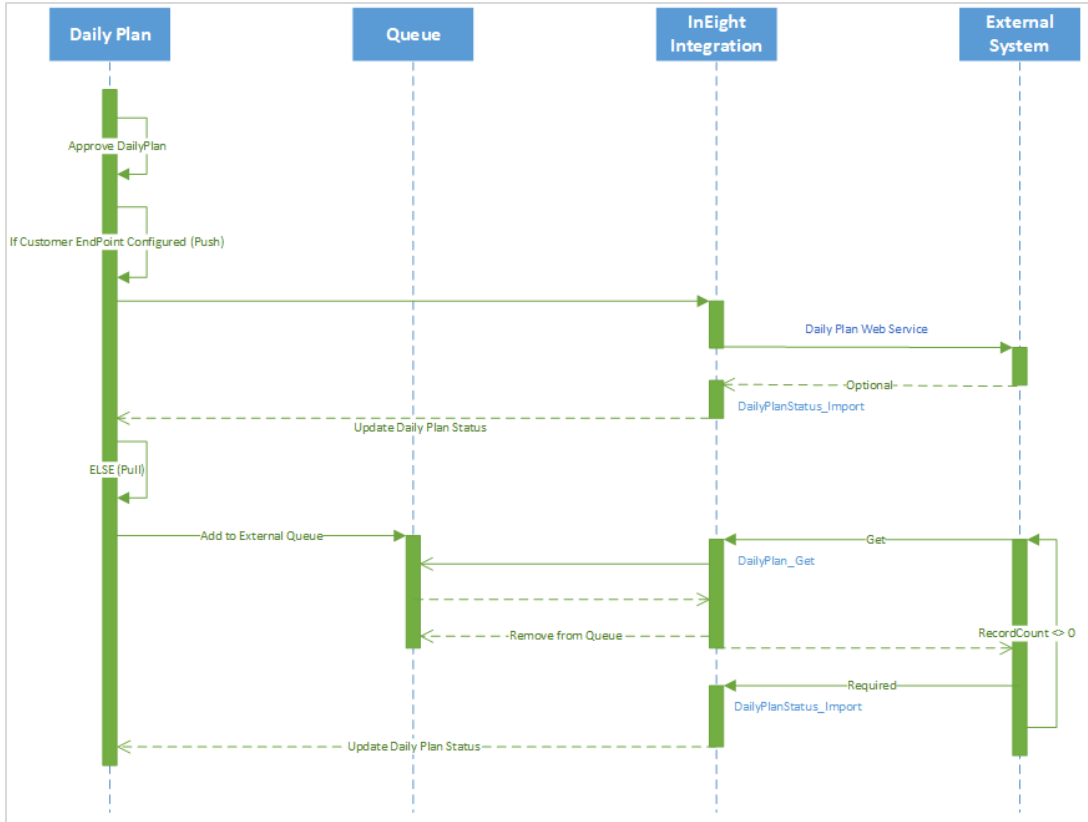
The recommended pattern for integration is the pull, which ensures integrity of the transaction, prevents failures during customer system down time, and allows the external system to determine when data is needed by either scheduled or manual processes. Timing of pull requests can be configured as quickly as one minute apart to get to near-real-time integration. This also provides more security because the push method requires opening a public URI or creating controlled network pathing to an end point that can be reached by the InEight cloud platform application. Also, the push method only supports a basic authentication with a pre-defined user name and password that must be passed each time.

**NOTE:**

Records that are placed in the external queue will be cleared after they are published to external system. Success confirmation will only update Daily Plan status to Final Processing Complete.

When retrieving time cards from the queue, a maximum of 200 records can be returned per request. If more than 200 records are available in the queue, the external system should confirm receipt of the 200 records returned and clear them from the queue, then make another request. The request/confirm cycle, as shown in the following flow diagram, should continue until the entire queue is cleared and no records are returned from the InEight cloud platform integration.





## Resynchronizing Daily Plans

Daily plans that have not successfully been pushed to or obtained by an external system can be re-synchronized when the daily plan is approved, and payroll sync status is either success or fail.

The image shows several daily plans that exist in a status that can be resynchronized with an external system. To perform the resynchronization, select one or more records from the UI, and then select the **Resync** icon on the toolbar.

Plan name	Shift	Location	Work package ID	Status	Status (bar)	Payroll Sync	Revision #
T01	First shift			Approved	<input checked="" type="checkbox"/>	Fail	0
test.01	First shift			Approved	<input checked="" type="checkbox"/>	Fail	0
Testhours	First shift			Approved	<input checked="" type="checkbox"/>		0
Case23	First shift			Approved	<input checked="" type="checkbox"/>		0
Offline				Approved	<input checked="" type="checkbox"/>		0
Test.log				Approved	<input checked="" type="checkbox"/>		0
Ex1efefafaf				Approved	<input checked="" type="checkbox"/>		0
Plan3				Approved	<input checked="" type="checkbox"/>		0
Test.Plan				Approved	<input checked="" type="checkbox"/>		0
ddddd				Approved	<input checked="" type="checkbox"/>	Fail	0

## Daily Plan Push

**NOTE:**

When using the push method, security is based on simple user and password credentials being supplied within the configuration. More advanced methods of authorization are not currently available.

Automatic pushes of daily plans can be configured in InEight Plan and InEight Progress applications. Consult with your implementation specialist for assistance on how to create the configuration in InEight cloud platform.

For this integration, there must be an available end point that can receive the data using a standard POST message with a JSON payload.

<b>Direction</b>	From InEight Plan	
<b>Frequency</b>	Each time a daily plan is set to a status of Approved it is immediately sent.	
<b>Trigger Methods</b>	Approval of a daily plan in InEight Plan.	
<b>Average Payload Size</b>	Each daily plan payload contains all details of the approved daily plan, resulting in tens or hundreds of records depending on how many employees and equipment associations exist.	
<b>APIM Name</b>	Not available in APIM	
<b>InEight Application</b>	<b>Starting Version</b>	18.1
	<b>Ending Version</b>	

Details for the data payload are provided in the [Fields](#), [Field Descriptions](#), and [Sample JSON](#) sections.

## Daily Plan Pull

To avoid gateway timeout issues, the API will return a maximum of  $n$  records available in the queue. The maximum of  $n$  is a configurable value in InEight cloud platform for each customer implementation and can hold a value up to 20. It is recommended for the integrating system to continue to call the API repeatedly until the record count is less than the value of  $n$  or until there are zero records returned. This will also clear the daily plans from the InEight system published to the external system.

Additionally, when using this method to obtain daily plans, it is required to provide confirmation of receipt and processing of records using the Daily Plan Status integration. This allows updating the Daily Plan status and Payroll Sync status.

<b>Direction</b>	From external system.	
<b>Frequency</b>	Determined by external system.	
<b>Trigger Methods</b>	Determined by external system.	
<b>Average Payload Size</b>	Each daily plan payload contains all details of approved daily plans that have been queued. Each daily plan can consist of tens or hundreds of records depending on how many employees and equipment associations exist. Therefore, each Get request can result in a large volume of data. A maximum of 20 records are returned with each request.	
<b>APIM Name</b>	DailyPlan_Get	
<b>InEight Application</b>	<b>Starting Version</b>	18.1
	<b>Ending Version</b>	

## Supported Filters

Data provided by the InEight cloud platform to external systems (outbound), can support selective fetching of data by applying filters in the API request.

Filter Name	Data Type	Description
JobCode	String	Optional: Returns records matching a project SourceSystemId with the value provided in this filter. Applying this filter affects performance of the integration. Therefore, it is recommended to retrieve all daily plans across all projects, and then have the receiving system apply filtering as needed in the business logic of that system.
MessageCount	Integer	Optional: Format: Int32 Specifies the number of records to return for the request. The maximum allowed value for this filter is 20.
CompleteMessage	Integer	<i>Not currently supported</i>

## Fields

This integration contains details of all items of a daily plan. In the payload, data is categorized into different sections that can be repeated for each daily plan and contain one or more sub-sections. The main categories are:

- **Daily Plan Detail:** Array of Daily Plans that match the criteria for being returned or sent in each integration request. The direct attributes of a Daily Plan record. All other sections are contained within this array.
  - [DailyPlanCostItem](#): Split of resources, hours, and codes against cost items used in the daily plan.
  - [DailyPlanNote](#): Note tags that have been assigned to resources or cost items on the daily plan.
  - [DailyPlanSignoff](#): Employee associated details and hours signed off against the daily plan.
  - [DailyPlanSignin](#): Employee associated details and hours signed in against the daily plan.
  - [DailyPlanBreaks](#): Employee associated details and breaks against the daily plan.
  - [DailyPlanBreakDetails](#): Details for crew breaks for the daily plan.
  - [EmployeeLevelBreaks](#): Details for employee breaks for the daily plan
  - [Executors](#): List of users and their roles that performed the function of executor for the daily plan.
  - [Approvers](#): List of users and their roles that performed the function of approver for the daily plan.
  - [DailyPlanEmployeePayrollIndicator](#): Detailed breakdown of the assignment of all payroll indicators to resources or cost items in the daily plan.
  - [DailyPlanClientSignoff](#): List of clients who signed off on the daily plan (if this functionality is enabled).
  - [DailyPlanMaintenance](#): Detailed breakdown of the use of resources when a maintenance work order or maintenance task is applied in a daily plan.
  - [ShiftDetails](#): Attribute information about the shift applied to the daily plan.

### Daily Plan Detail

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	DailyPlan	Array		
2	<a href="#">DailyPlanCostItem</a>	Array		DailyPlan
2	<a href="#">DailyPlanNote</a>	Array		DailyPlan
2	<a href="#">DailyPlanSignoff</a>	Array		DailyPlan
2	<a href="#">Executors</a>	Array		DailyPlan
2	<a href="#">Approvers</a>	Array		DailyPlan
2	<a href="#">DailyPlanEmployeePayrollIndicator</a>	Array		DailyPlan
2	<a href="#">DailyPlanClientSignoff</a>	Array		DailyPlan
2	PlanId <sup>2</sup>	String	50	DailyPlan
2	CreateDate <sup>3</sup>	String	25	DailyPlan

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	ProjectId	String	50	DailyPlan
2	LanguageKey	String	2	DailyPlan
2	Location	String	100	DailyPlan
2	CreatedById	String	50	DailyPlan
2	DeviceKey	String	50	DailyPlan
2	Shift	String	100	DailyPlan
2	PlanDate <sup>3</sup>	String	25	DailyPlan
2	PlanTitle	String	100	DailyPlan
2	PlanStatusCode	String	100	DailyPlan
2	ModifiedById	String	50	DailyPlan
2	ModifiedDate <sup>3</sup>	String	25	DailyPlan
2	PlannedDate <sup>3</sup>	String	25	DailyPlan
2	PlannedById	String	50	DailyPlan
2	ExecutedDate <sup>3</sup>	String	25	DailyPlan
2	ExecutedById	String	50	DailyPlan
2	ApprovedDate <sup>3</sup>	String	25	DailyPlan
2	ApprovedById	String	50	DailyPlan
2	ErrorHandlingToken	String	50	DailyPlan
2	CommitmentCode	String	100	DailyPlan
2	ExternalSyncStatus	Number	Integer	DailyPlan
2	CommitmentId	String	Integer	DailyPlan
2	<a href="#">DailyPlanMaintenance</a>	Array		DailyPlan
2	<a href="#">ShiftDetails</a>	Array		DailyPlan
2	ClientSignoffReportRecipients	List	1000	DailyPlan

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.

## Daily Plan Cost Item

This array and all sub-arrays will be repeated for each cost item used in the daily plan.

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanCostItem	Array		DailyPlan
3	DailyPlanCostItemEmployee	Array		DailyPlanCostItem
4	ResourceId	String	50	DailyPlanCostItemEmployee
4	SubmittedDoubletimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	SubmittedOvertimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	SubmittedStandardHours	Decimal	5,2	DailyPlanCostItemEmployee
4	ApprovedDoubletimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	ApprovedOvertimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	ApprovedStandardHours	Decimal	5,2	DailyPlanCostItemEmployee
4	PlannedDoubletimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	PlannedOvertimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	PlannedStandardHours	Decimal	5,2	DailyPlanCostItemEmployee
4	EmployeeId	String	50	DailyPlanCostItemEmployee
4	CraftCode	String	50	DailyPlanCostItemEmployee
4	BillingClass	String	50	DailyPlanCostItemEmployee
4	ReasonCodes	Array		DailyPlanCostItemEmployee
5	ReasonCodeId	String	50	ReasonCodes
5	Hour Type	String	50	ReasonCodes
5	Hours	Decimal	5,2	ReasonCodes
5	BillingCode	String	10	ReasonCodes
5	Premiums	List		ReasonCodes
3	DailyPlanCostItemEquipment	Array		DailyPlanCostItem
4	ResourceId	String	50	DailyPlanCostItemEquipment
4	ApprovedStandardHours	Decimal	5,2	DailyPlanCostItemEquipment
4	PlannedStandardHours	Decimal	5,2	DailyPlanCostItemEquipment
4	SubmittedStandardHours	Decimal	5,2	DailyPlanCostItemEquipment
4	EquipmentId	String	50	DailyPlanCostItemEquipment
4	EquipmentType	String	250	DailyPlanCostItemEquipment
4	ReasonCodes	Array		DailyPlanCostItemEquipment

Depth	Name	Type	Precision <sup>1</sup>	Parent
5	OperatedEmployeeId	String	50	ReasonCodes
5	ReasonCodeId	String	50	ReasonCodes
5	Hours	Decimal	5,2	ReasonCodes
5	BillingCode	String	10	ReasonCodes
3	DailyPlanCostItemComponent	Array		DailyPlanCostItem
4	ResourceId	String	50	DailyPlanCostItemComponent
4	ComponentId	Number	50	DailyPlanCostItemComponent
4	ProjectEstimatingResourceId	Number	50	DailyPlanCostItemComponent
4	ProjectEstimatingResourceSourceSystemId	String	50	DailyPlanCostItemComponent
4	InstalledQuantity	Decimal	28,15	DailyPlanCostItemComponent
4	ComponentType	String	50	DailyPlanCostItemComponent
3	DailyPlanCostItemVendors	Array	NA	DailyPlanCostItem
4	VendorId	Number	BigInt	DailyPlanCostItemVendors
4	VendorDisplay	String	250	DailyPlanCostItemVendors
4	SubmittedDoubletimeHours	Decimal	5,2	DailyPlanCostItemVendors
4	SubmittedOvertimeHours	Decimal	5,2	DailyPlanCostItemVendors
4	SubmittedStandardHours	Decimal	5,2	DailyPlanCostItemVendors
4	ApprovedDoubletimeHours	Decimal	5,2	DailyPlanCostItemVendors
4	ApprovedOvertimeHours	Decimal	5,2	DailyPlanCostItemVendors
4	ApprovedStandardHours	Decimal	5,2	DailyPlanCostItemVendors
4	PlannedDoubletimeHours	Decimal	5,2	DailyPlanCostItemVendors
4	PlannedOvertimeHours	Decimal	5,2	DailyPlanCostItemVendors
4	PlannedStandardHours	Decimal	5,2	DailyPlanCostItemVendors
4	PlannedNumberOfEmployees	Number	BigInt	DailyPlanCostItemVendors
4	SubmittedNumberOfEmployees	Number	BigInt	DailyPlanCostItemVendors
4	ApprovedNumberOfEmployees	Number	BigInt	DailyPlanCostItemVendors
4	ReasonCodes	Number	BigInt	DailyPlanCostItemVendors
5	ReasonCode	Array	NA	ReasonCodes
5	HourType	String	50	ReasonCodes
5	Hours	Decimal	16,5	ReasonCodes
3	CostItemId	String	50	DailyPlanCostItem

Depth	Name	Type	Precision <sup>1</sup>	Parent
3	TaskPriority	String	50	DailyPlanCostItem
3	PlanQuantity	Decimal	28,15	DailyPlanCostItem
3	SubmittedQuantity	Decimal	28,15	DailyPlanCostItem
3	ApprovedQuantity	Decimal	28,15	DailyPlanCostItem
3	CostItemSourceSystemId	String	50	DailyPlanCostItem
3	WBSCode	String	50	DailyPlanCostItem

### Daily Plan Note

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanNote	Array		DailyPlan
3	DailyPlanNoteEmployee	Array		DailyPlanNote
4	EmployeeDisplay	String	50	DailyPlanNoteEmployee
3	DailyPlanNoteEquipment	Array		DailyPlanNote
4	EquipmentId	String	50	DailyPlanNoteEquipment
3	DailyPlanNoteTag	Array		DailyPlanNote
4	DailyPlanNoteTagId	Number	BigInt	DailyPlanNoteTag
4	TagCode	String	100	DailyPlanNoteTag
3	DailyPlanNoteTask	Array		DailyPlanNote
4	CostItemId	Number	50	DailyPlanNoteTask
3	DailyPlanNoteMaintenance	Array		DailyPlanNote
4	MaintenanceEquipmentID	String	50	DailyPlanNoteMaintenance
4	WBSPHaseCode	String	50	DailyPlanNoteMaintenance
4	WorkOrderID	String	50	DailyPlanNoteMaintenance
3	DailyPlanNoteVendors	Array		DailyPlanNote
4	VendorDisplay	String	250	DailyPlanNoteVendors
3	Noteld	String	50	DailyPlanNote
3	Description	String	4000	DailyPlanNote



## Daily Plan Signoff

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanSignoff	Array		DailyPlan
3	EmployeeId	String	50	DailyPlanSignoff
3	QuestionnaireKey	String	50	DailyPlanSignoff
3	SignoffEmployeeId	String	50	DailyPlanSignoff
3	SignoffDate <sup>3</sup>	String	25	DailyPlanSignoff
3	SignoffStandardHours	Decimal	5,2	DailyPlanSignoff
3	SignoffOvertimeHours	Decimal	5,2	DailyPlanSignoff
3	SignoffDoubletimeHours	Decimal	5,2	DailyPlanSignoff
3	DailyPlanSignoffResponse	Array		DailyPlanSignoff
4	EmployeeId	String	50	DailyPlanSignoffResponse
4	QuestionKey	String	50	DailyPlanSignoffResponse
4	DailyPlanSignoffResponseId	String	50	DailyPlanSignoffResponse
4	ResponseKey	String	50	DailyPlanSignoffResponse
4	ResponseText	String	50	DailyPlanSignoffResponse

## Daily Plan Signin

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanSignins	Array		DailyPlan
3	EmployeeDisplay	String	50	DailyPlanSignins
3	SignedInByEmployeeDisplay	String	50	DailyPlanSignins
3	SignedInByUserDisplay	String	50	DailyPlanSignins
3	SignedInDate	String	25	DailyPlanSignins
4	DailyPlanSignInResponses	Array		DailyPlanSignins
4	DailyPlanSignInResponseId	String	50	DailyPlanSignInResponses
4	QuestionKey	String	50	DailyPlanSignInResponses
4	QuestionDescription	String	50	DailyPlanSignInResponses
4	ResponseKey	String	50	DailyPlanSignInResponses
4	ResponseDescription	String	50	DailyPlanSignInResponses

### Daily Plan Breaks

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanBreaks	Array		DailyPlan
3	EmployeeDisplay	String	50	DailyPlanBreaks
3	BreakNumber	Number	50	DailyPlanBreaks
3	BreakStartTime	String	25	DailyPlanBreaks
3	BreakInMinutes	Number	50	DailyPlanBreaks

### Daily Plan Break Details

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanBreakDetails	Array		DailyPlan
3	DailyPlanLevelBreaks	Array		DailyPlanBreakDetails
	BreakNumber	Number	50	DailyPlanBreaks
3	BreakStartTime	String	25	DailyPlanBreaks
3	BreakInMinutes	Number	50	DailyPlanBreaks

### Employee Level Breaks

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	EmployeeLevelBreaks	Array		DailyPlan
3	EmployeeDisplay	String	50	DailyPlanBreaks
3	BreakNumber	Number	50	DailyPlanBreaks
3	BreakStartTime	String	25	DailyPlanBreaks
3	BreakInMinutes	Number	50	DailyPlanBreaks

### Executors

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	Executors	Array		DailyPlan
3	RoleName	String	100	Executors
3	DisplayId	String	50	Executors
4	ExecutorPosition	Number	50	Executors

## Approvers

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	Approvers	Array		DailyPlan
3	RoleName	String	100	Approvers
3	DisplayId	String	50	Approvers
4	ApproverPosition	Number	50	Approvers

## Daily Plan Employee Payroll Indicator

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanEmployeePayrollIndicator	Array		DailyPlan
3	DailyPlanEmployeePayrollIndicatorId	Number	50	DailyPlanEmployeePayrollIndicator
3	EmployeeId	String	50	DailyPlanEmployeePayrollIndicator
3	PayrollIndicatorId	Number	50	DailyPlanEmployeePayrollIndicator
3	DailyPlanNotId	Number	50	DailyPlanEmployeePayrollIndicator
3	CostItemId	Number	50	DailyPlanEmployeePayrollIndicator
3	PayrollIdentifier	String	50	DailyPlanEmployeePayrollIndicator
3	WBSPhaseCode	String	50	DailyPlanEmployeePayrollIndicator
3	EmployeeSourceSystemId	String	50	DailyPlanEmployeePayrollIndicator

## Daily Plan Client Signoff

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanClientSignoff	Array		DailyPlan
3	Name	String	100	ClientSignoff
3	Email	String	100	ClientSignoff

## Daily Plan Maintenance

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	DailyPlanMaintenance	Array		DailyPlan
3	EmployeeId	String	50	DailyPlanMaintenance
3	WorkOrderId	String	100	DailyPlanMaintenance
3	WBSPhaseCode	String	50	DailyPlanMaintenance
3	CostItemSourceSystemId	String	50	DailyPlanMaintenance

Depth	Name	Type	Precision <sup>1</sup>	Parent
3	TotalHours	Decimal	16,5	DailyPlanMaintenance
3	MaintenanceEquipmentID	String	50	DailyPlanMaintenance
3	Segment1	String	50	DailyPlanMaintenance
3	Segment2	String	50	DailyPlanMaintenance
3	Segment3	String	50	DailyPlanMaintenance
3	Segment4	String	50	DailyPlanMaintenance
3	ReasonCodes	Array		DailyPlanMaintenance
4	ReasonCodeId	String	50	ReasonCodes
4	Hour Type	String	50	ReasonCodes
4	Hours	Decimal	5,2	ReasonCodes
4	Premiums	List		ReasonCodes

### Shift Details

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	ShiftDetails	Array		DailyPlan
3	Shift	String	100	ShiftDetails
3	ShiftStartDateTime <sup>3</sup>	String	25	ShiftDetails
3	ShiftEndDateTime <sup>3</sup>	String	25	ShiftDetails
3	EmployeeShiftDetails	Array		ShiftDetails
4	EmployeeId	String	50	
4	ShiftStartDateTime	String	25	
4	ShiftEndDateTime	String	25	

### Field Descriptions

This integration has distinct grouping of data sets with some duplication of fields across them. To clarify the information and it more understandable, each data set is described independently.

**NOTE:** The order of fields in the Field Descriptions table does not match the order in the JSON provided with the API request. The attributes of each group are at the top of the section instead of appearing after sub-sections as they are in the JSON.

Name	Description	Example
DailyPlan	Array, this entire data set is repeated for each daily plan returned in the results.	

Name	Description	Example
PlanId	InEight ID of the daily plan.	92723
CreatedDate	Date when the daily plan was originally created.	2017:04:23T12:17:26
ProjectId	DisplayId of the project associated with the daily plan.	105012
LanguageKey	DisplayId of the default language of the daily plan.	EN
Location	Free-form text that describes the location of the project where the work will be performed.	South Bridge
CreatedById	DisplayId of the user who created the daily plan.	135464
DeviceKey	GUID of a device (usually mobile) on which the daily plan was created.	5870e373-594b-4321-8831-99106728ddc8
Shift	Code that represents the shift when the work was performed.	01
PlanDate	Date the daily plan will be or was executed. This is the date that represents the hours used for payroll.	2017:05:23T12:00:00
PlanTitle	Free-form text to describe the name of the daily plan.	Component
PlanStatusCode	<p>Current status of the daily plan. The following are the possible status codes of a daily plan:</p> <ul style="list-style-type: none"> <li>• PLAN - Planning phase</li> <li>• EXEC - Execution phase</li> <li>• WAPP - Waiting for approval</li> <li>• APPR - Approved</li> <li>• FINL - Final, closed</li> </ul> <p>Only daily plans with a status of APPR, indicating they have been approved, are made available to external systems.</p>	APPR
ModifiedById	ID of a user that modified the daily plan.	6543134
ModifiedDate	Date of the last modification to the daily plan.	2017:05:23T03:25:30
PlannedDate	Date on which the daily plan was moved into the planning phase.	2017:05:23T03:25:30
PlannedById	ID of the user that moved the daily plan into the planning phase.	231344
ExecutedDate	Date on which the daily plan was moved into execution phase.	2017:05:23T03:25:30
ExecutedById	ID of the User that moved the daily plan into the execution phase.	231344
ApprovedDate	Date on which the daily plan was moved into the approval phase.	2017:05:23T03:25:30
ApprovedById	ID of the User that moved the daily plan into the approval phase.	231344

Name	Description	Example
ErrorHandlingToken	Unique token that can be used to either retrieve or report issues related to the integration instance.	5870e373-594b-4321-8831-99106728ddc8
CommitmentCode	Display number of the commitment/purchase order that was assigned to the daily plan.	11233455
ExternalSyncStatus	Allowed values are: <ul style="list-style-type: none"> <li>• 0 = Ready to send</li> <li>• 1 = Success</li> <li>• 2 = FAIL</li> </ul>	1
CommitmentId	InEight ID for a commitment/purchase order assigned to the daily plan.	
ClientSignOffReportRecipients	List of email addresses of people to receive a copy of the Client Signoff Report.	john.smith@bigco.com bill.bob@bigco.com karen.kim@bigco.com mary.smith@bigcon.com

<b>DailyPlanCostItem</b> – (main section) For each cost item (task in UI) assigned to the daily plan, details about how resources (Employees, Equipment, components) were used in the operation are broken out. This main section and its sub-sections are repeated for each cost item assigned to the daily plan.		
DailyPlanCostItem	Array header for cost item.	
CostItemId	InEight ID of the cost item associated to the daily plan.	142712
TaskPriority	Order of tasks in the daily plan.	1
PlanQuantity	Quantity planned for the day to complete.	300
SubmittedQuantity	Quantity entered/submitted by the foreman.	275
ApprovedQuantity	Quantity entered/approved.	275
CostItemSourceSystemId	Unique ID of the cost item from the source system of cost items. For example, if cost items are imported from an external system/ERP, the ID from that system would in this field. This is for referential mapping.	4464444
WBSCode	WBS phase code for the given cost item indicated by CostItemSourceSystemId	46577

<b>DailyPlanCostItemEmployee</b> – (subsection) Contains details for each combination of cost item and employee assigned to the daily plan. This section is repeated for each employee and cost item association.		
DailyPlanCostItemEmployee	Array header for employee.	

ResourceId	DisplayId (Employee ID in the UI) of the employee associated to the daily plan cost item. There can be many employees for each cost item.	333903
SubmittedDoubletimeHours	Double time hours submitted by employee.	0.00
SubmittedOvertimeHours	Overtime hours submitted by employee.	1.00
SubmittedStandardHours	Straight time hours submitted by employee.	8.00
ApprovedDoubletimeHours	Double time hours approved for employee and used for payroll processing.	0.00
ApprovedOvertimeHours	Overtime hours approved for employee and used for payroll processing.	1.00
ApprovedStandardHours	Straight time hours approved for employee and used for payroll processing.	8.00
PlannedDoubletimeHours	Double time hours planned for employee.	0.00
PlannedOvertimeHours	Overtime hours planned for employee.	0.00
PlannedStandardHours	Straight time planned for employee.	8.00
EmployeeId	Internal InEight ID for the Employee.	333903
CraftCode	DisplayId of a valid craft in InEight master data.	Carpenter
BillingClass	Billing class that has been applied to the employee for this plan.	Operator 1
ReasonCodes	Array header for reason codes associated to the employee/cost item combination.	
ReasonCodeId	DisplayId of a reason code.	OPT
Hour Type	Description of the Hour Type pay scale associated with the reason and hours	Standard Time, Overtime, Double Time
Hours	Number of hours associated to the employee/cost item/reason code.	1.25
BillingCode	Code used to associate with the InEight Billings module.	ST01
Premiums	A comma separated list of premium codes associated with the employee/cost item/reason code.	"SA", "LOC", "HEIGHT"

<b>DailyPlanCostItemEquipment</b> – (subsection) Contains details for each combination of cost item and equipment assigned to the daily plan. This section is repeated for each equipment and cost item association.		
DailyPlanCostItemEquipment	Array header for equipment.	
ResourceId	DisplayId (Equipment Id in the UI) for each piece of equipment on the timesheet. There can be multiple records for each cost items.	ETL

ApprovedStandardHours	Hours approved for equipment and used for final equipment hours.	6.00
PlannedStandardHours	Planned equipment hours.	6.00
SubmittedStandardHours	Hours entered against the equipment.	6.00
EquipmentId	Internal InEight ID for the equipment record.	1154988
EquipmentType	Equipment type that corresponds to the equipment.	Crane
ReasonCodes	Array header for reason codes associated to the equipment/cost item.	
OperatedEmployeeId	Display ID of the employee, if any, that operated the equipment.	113444
ReasonCodeId	Abbreviated code for a specific reason.	OPT
Hours	Number of hours associated to the equipment/cost item/reason code.	1.25
BillingCode	Code used to associate with the InEight Billings module.	EQST01

**DailyPlanCostItemComponent** – (subsection) Contains details for each component used in a cost item assigned to the daily plan. Data is only populated in this array when the component used in a daily plan has an Estimating Material (Component Details > Procurement section) that also has an association to a cost item. This section will repeat for each component associated to the cost item that meets the criteria above.

DailyPlanCostItemComponent	Array header for component.	
ResourceId	ID of the estimating resource that is displayed in the applications.	479873
ComponentId	Internal InEight ID from InEight Plan for the component.	123
ProjectEstimatingResourceId	ID of the estimating resource, which would be the material.	123
ProjectEstimatingResourceSourceSystemId	ID of the estimating resource in the source system.	7903
InstalledQuantity	Quantity of work claimed against the component.	12353.343
ComponentType	Component type of the component. Possible values are: <ul style="list-style-type: none"> <li>• Material</li> <li>• Activity.</li> </ul>	Activity

**DailyPlanCostItemVendors** – (subsection) Details for each vendor associated to a cost item in this section. This section is repeated for each cost item.

DailyPlanCostItemVendors	Array header for the vendors.	NA
--------------------------	-------------------------------	----



VendorId	Unique ID of the estimating resource that shows in the UI.	113
VendorDisplay	Display name of associated vendor	0010106485
SubmittedDoubletimeHours	Double time hours submitted by vendor.	0.00
SubmittedOvertimeHours	Overtime time hours submitted by vendor.	1.00
SubmittedStandardHours	Straight time hours submitted by vendor.	8.00
ApprovedDoubletimeHours	Double time hours approved for vendor.	0.00
ApprovedOvertimeHours	Overtime time hours approved for vendor.	1.00
ApprovedStandardHours	Straight time hours approved for vendor.	8.00
PlannedDoubletimeHours	Double time hours planned for vendor.	0.00
PlannedOvertimeHours	Overtime time hours planned for vendor.	0.00
PlannedStandardHours	Straight time hours planned for vendor.	8.00
PlannedNumberOfEmployees	Planned number of resources for vendor.	1
SubmittedNumberOfEmployees	Submitted number of resources for vendor.	1
ApprovedNumberOfEmployees	Approved number of resources for vendor.	1
ReasonCodes	Array header for reason codes associated to the employee/cost item combination.	
ReasonCode		
HourType	Description of the Hour Type pay scale associated with the reason and hours	

<b>DailyPlanNote</b> – (main section) Details for each note associated to a daily plan are in this section. This section is repeated for each note.		
DailyPlanNote	Array header for the note.	NA
DailyPlanNoteEmployee	Array header for the employee note.	NA
EmployeeDisplay	Employee associated with the note. There can be many employees associated with a note.	333903
DailyPlanNoteEquipment	Array header for the equipment note.	NA
EquipmentId	InEight ID for the equipment associated with the note. There can be many equipment records associated to a note.	1154988
DailyPlanNoteTag	Array header for section for all note tags assigned to the daily plan. This section repeats for each note tag record.	NA

TagCode	Tag code associated with the note. Tag codes are derived from a list provided to InEight Plan from a payroll or HR system. There can be many tag codes associated with a note.	STD_1
DailyPlanNoteTask	Array header for section for all note tasks assigned to the daily plan. This section repeats for each note task record.	NA
CostItemId	Display ID of a cost item associated with the note. There can be many cost item records for a note.	142712
DailyPlanNoteMaintenance	Array header for section for all note maintenance assigned to the daily plan. This section repeats for each note task record.	NA
MaintenanceEquipmentId	Source System ID of the associated equipment.	142732
WBSPHaseCode	WBS phase code of the associated cost item.	144232
WorkOrderId	SourceSystem ID of work order	342345
DailyPlanNoteVendors		
VendorDisplay		0010106485
NotelId	Internal identifier for a specific note.	37646
Description	Free-form text of the note.	No accidents reported for day

<b>DailyPlanSignoff – (main section) This section only appears one time for each daily plan.</b>		
DailyPlanSignoff	Array header for the signoff of the daily plan.	NA
EmployeeId	InEight ID of the employee maintained by the InEight cloud platform. This is not the Display Id that users would view in the UI.	333903
QuestionnaireKey	Code that represents the questions asked when signing-off hours.	2000
SignoffEmployeeId	Employee who performed the signoff. This can be different from the Employee ID for the hours that are being signed off against.	333903
SignoffDate	Date of the signoff.	2017:05:23T12:17:26
SignoffStandardHours	Submitted hours that are being signed off against the daily plan.	8.00
SignoffOvertimeHours	Submitted hours that are being signed off against the daily plan.	1.00
SignoffDoubletimeHours	Submitted hours that are being signed off against the daily plan.	0.00

DailyPlanSignoffResponse	Array header for the signoff response. This section is repeated for each employee that signed off for their hours.	NA
EmployeeId	Employee associated with the signoff records. There can be multiple signoff records per daily plan.	333903
QuestionKey	Display value/code that is used to identify a question in the signoff questionnaire. Each question has a unique code assigned by an ERP to identify the question. For example, 2001 could represent the question "I was injured today." which is shown in the signoff screen.	2001
DailyPlanSignoffResponseId	Unique internal InEight ID for the responses submitted for the signoff questionnaire on that daily plan.	225466
ResponseKey	Display value/code that could be used to identify a unique response. The value represents a code that could be used by an ERP to identify a YES/NO response to any of the signoff questions.	20120
ResponseText	Actual text a user views in the UI for a response to a question.	No

<b>DailyPlanSignin</b> – (main section) This section only appears one time for each daily plan.		
EmployeeDisplay	Display Id of the employee being signed in.	333903
SignedInByEmployeeDisplay	Display ID of the employee performing the sign in.	333903
SignedInByUserDisplay	Display ID of the logged in user performing the sign in or using on behalf of signout all eligible option (this value exists only if logged in user does not have association employee).	333563
SignedInDate	Date and time when the employee was signed in.	2021/02/04 13:26:57
DailyPlanSignInResponses	Array header for Signin response.	
DailyPlanSignInResponseId	Unique internal InEight ID for the responses submitted for the sign in questionnaire on that daily plan.	6
QuestionKey	Display value/code that is used to identify a question in the sign in questionnaire. Each question has a unique code assigned by an ERP to identify the question. For example, 2001 could represent the question "I was injured today." which is shown in the signoff screen.	1
QuestionDescription	Full text of the signin question.	I have received today's safety briefing

ResponseKey	Display value/code that could be used to identify a unique response. The value represents a code that could be used by an ERP to identify a YES/NO response to any of the signoff questions.	1
ResponseDescription	Actual text a user sees in the UI for a response to a question.	Yes

<b>DailyPlanBreaks</b> – (main section) This section only appears one time for each daily plan.		
EmployeeDisplay	Display ID of the employee taking a break.	333903
BreakNumber	Break number for the day. There can be multiple breaks per day.	1
BreakStartTime	Employee break start date and time.	2021/02/04 00:02:00
BreakInMinutes	Duration of the break in minutes.	15

<b>DailyPlanBreakDetail</b> – (main section) This section only appears one time for each daily plan.		
DailyPlanLevelBreaks	Array header.	
BreakNumber	Break number for the day. There can be multiple breaks per day.	1
BreakStartTime	Employee break start date and time.	2021/02/04 00:02:00
BreakInMinutes	Duration of the break in minutes.	15

<b>EmployeeLevelBreaks</b> – (main section) This section only appears one time for each daily plan.		
EmployeeDisplay	Display ID of the employee taking a break.	333903
BreakNumber	Break number for the day. There can be multiple breaks per day.	1
BreakStartTime	Employee break start date and time.	2021/02/04 00:02:00
BreakInMinutes	Duration of the break in minutes.	15

<b>Executers</b> – (main section) This section only appears one time for each daily plan.		
Executers	Array header for the executers of the daily plan.	NA
RoleName	User role selected to represent the executor of the daily plan.	Foreman

DisplayId	Display ID of the user that executed the daily plan.	104017
ExecutorPosition	Position in the UI where the executor is added	1

<b>Approvers</b> – (main section) This section only appears one time for each daily plan.		
Approvers	Array header for the approvers of the daily plan.	NA
DisplayId	Display ID of the user that approved the daily plan.	104023
RoleName	User role selected to represent the approver of the daily plan.	Engineer
ApproverPosition	Position in the UI where the approver is added	1

<b>DailyPlanEmployeePayrollIndicator</b> – (main section) This section only appears one time for each daily plan.		
DailyPlanEmployeePayrollIndicator	Array header for the employee payroll indicators in the daily plan.	NA
DailyPlanEmployeePayrollIndicatorId	Internal InEight ID for the payroll indicator that was assigned to the employee on a daily plan.	7
EmployeeId	Employee display ID for who the payroll indicator was assigned.	384683
PayrollIndicatorId	Internal InEight ID of the payroll indicator.	12
DailyPlanNoteId	Internal InEight ID for the note, if a note was created and generated for the payroll indicator assignment.	432
CostItemId	Cost item against which the payroll indicator was assigned to on a daily plan.	23
PayrollIdentifier	Display ID or code for the payroll indicator that was assigned.	FIN, PAY, FOD
WBSPhaseCode	WBS phase code for the cost item that the payroll indicator was assigned.	1011
EmployeeSourceSystemId	SourceSystemId of the Employee that the indicator was applied to.	2313444

<b>DailyPlanClientSignoff</b> – (main section) This section only appears one time for each daily plan.		
DailyPlanClientSignOff	Array header for the client that signed off the daily plan.	NA
Name	Name of the client that signed off on the daily plan. This is an optional configuration item.	John Smith
Email	Email address of the client who signed off on the daily plan.	john.smith@bigco.com

<p><b>DailyPlanMaintenance</b> – (main section) Multiple work orders can be recorded for a single daily plan, so several records might be included in the array for this data section.</p> <p>Time recorded for maintenance can be against a work order (see <a href="#">Daily Plan Work Orders</a>), or against a cost item. When recorded against a work order, typically there will not be an associated cost item. The reverse is also true; when time is recorded directly against a cost item, typically there will not be an associated work order.</p>		
DailyPlanMaintenance	Array header for maintenance.	
EmployeeId	Display ID of the employee whose time was recorded for the work order.	1154854
WorkOrderId	Source System ID of the work order (see <a href="#">Daily Plan Work Orders</a> for details).	159116
WBSPhaseCode	WBS phase code of the associated cost item.	1540
CostItemSourceSystemId	Source System Id of the associated cost item.	654744
TotalHours	Hours associated to the cost item or work order. If there are multiple reason codes against the maintenance hours entered, this would contain the total of all hours entered against different reason codes, against that specific work order or cost item, and for the employee and equipment.	15.25
MaintenanceEquipmentID	Source System ID of the associated equipment.	6546765
Segment1	First segment of the cost item associated with the hours.	104354
Segment2	Second segment of the cost item associated with the hours.	104354A
Segment3	Third segment of the cost item associated with the hours.	4093
Segment4	Fourth segment of the cost item associated with the hours.	
ReasonCodes	Array header for reason codes associated to the equipment maintenance.	
ReasonCodeId	Abbreviated code for a specific reason.	SCHEDULED
Hour Type	Description of the Hour Type pay scale associated with the reason and hours	Standard Time, Overtime, Double Time
Hours	Number of hours associated to the equipment maintenance record.	.50
Premiums	Comma separated list of premium codes associated with the equipment maintenance/reason code.	HEIGHT, LOC

**ShiftDetails** – (main section) Multiple shifts can be recorded for a single daily plan, so several records can be included in the array for this data section. Additionally, there is an array for which employees worked in each shift.

Shift	The name of the shift	Swing Shift
ShiftStartDateTime	The start date and time of the shift	2020/02/26 06:00:00
ShiftEndDateTime	The end date and time of the shift	2020/02/27 06:00:00
EmployeeShiftDetails	Array of employees that worked in the shift on the daily plan	
EmployeeId	The Display Id of the employee that worked	5164677
ShiftStartDateTime	The start date and time of the shift	2020/02/26 06:00:00
ShiftEndDateTime	The end date and time of the shift	2020/02/27 06:00:00

## Sample JSON

### Example 1: Daily Plan with Reason Code hours

```
[{
  "PlanId": 2015903,
  "CreatedDate": "2022/06/23 21:02:23",
  "ProjectId": "105164",
  "LanguageKey": "EN",
  "Location": "",
  "CreatedById": "",
  "DeviceKey": "",
  "Shift": "01",
  "PlanDate": "2022/08/12 00:00:00",
  "PlanTitle": "today",
  "PlanStatusCode": "APPR",
  "ModifiedById": "",
  "ModifiedDate": "2022/08/12 15:44:48",
  "PlannedDate": "2022/06/23 21:03:28",
  "PlannedById": "",
  "ExecutedDate": "2022/08/12 10:42:12",
  "ExecutedById": "",
  "ApprovedDate": "2022/08/12 10:44:47",
  "ApprovedById": "",
  "ErrorHandlingToken": "",
  "ExternalSyncStatus": 2,
  "IsMailSent": false,
  "DailyPlanCostItem": [{
    "CostItemId": "1366679",
    "TaskPriority": 2,
    "PlanQuantity": 0.0,
    "SubmittedQuantity": 0.0,
    "ApprovedQuantity": 0.0,
    "DailyPlanCostItemEmployee": [{
      "ResourceId": "00493146",
      "SubmittedDoubletimeHours": "",
      "SubmittedOvertimeHours": "",
      "SubmittedStandardHours": "",
    }
  ]
}
```

```

        "ApprovedDoubletimeHours": "",
        "ApprovedOvertimeHours": "",
        "ApprovedStandardHours": "",
        "PlannedDoubletimeHours": "",
        "PlannedOvertimeHours": "",
        "PlannedStandardHours": "",
        "EmployeeId": 186739,
        "CraftCode": "OPJA",
        "BillingClass": "",
        "ReasonCodes": []
    }],
    "DailyPlanCostItemEquipment": [],
    "DailyPlanCostItemComponent": [],
    "DailyPlanCostItemVendors": [],
    "CostItemSourceSystemId": "0a37696365d2491cb4d39441c24dedd1",
    "WBSCode": "1009"
}, {
    "CostItemId": "1366680",
    "TaskPriority": 1,
    "PlanQuantity": 0.0,
    "SubmittedQuantity": 0.0,
    "ApprovedQuantity": 0.0,
    "DailyPlanCostItemEmployee": [{
        "ResourceId": "00493147",
        "SubmittedDoubletimeHours": 0.0,
        "SubmittedOvertimeHours": 0.0,
        "SubmittedStandardHours": 1.0,
        "ApprovedDoubletimeHours": 0.0,
        "ApprovedOvertimeHours": 0.0,
        "ApprovedStandardHours": 1.0,
        "PlannedDoubletimeHours": "",
        "PlannedOvertimeHours": "",
        "PlannedStandardHours": "",
        "EmployeeId": 186740,
        "CraftCode": "OPJA",
        "BillingClass": "",
        "ReasonCodes": [{
            "ReasonCodeId": "",
            "HourType": "Standard time",
            "Hours": 1.0,
            "BillingCode": "ST01"
        }],
        "Premiums": []
    }],
    "ResourceId": "00493233",
    "SubmittedDoubletimeHours": 0.0,
    "SubmittedOvertimeHours": 0.0,
    "SubmittedStandardHours": 1.0,
    "ApprovedDoubletimeHours": 0.0,
    "ApprovedOvertimeHours": 0.0,
    "ApprovedStandardHours": 1.0,
    "PlannedDoubletimeHours": "",
    "PlannedOvertimeHours": "",
    "PlannedStandardHours": "",
    "EmployeeId": 186846,
    "CraftCode": "ZWPJ",
    "BillingClass": "",

```



```

        "ReasonCodes": [{
            "ReasonCodeId": "",
            "HourType": "Standard time",
            "Hours": 1.0,
            "BillingCode": "ST01"
            "Premiums": []
        }]
    }, {
        "ResourceId": "00493146",
        "SubmittedDoubletimeHours": 0.0,
        "SubmittedOvertimeHours": 0.0,
        "SubmittedStandardHours": 1.0,
        "ApprovedDoubletimeHours": 0.0,
        "ApprovedOvertimeHours": 0.0,
        "ApprovedStandardHours": 1.0,
        "PlannedDoubletimeHours": "",
        "PlannedOvertimeHours": "",
        "PlannedStandardHours": "",
        "EmployeeId": 186739,
        "CraftCode": "OPJA",
        "BillingClass": "",
        "ReasonCodes": [{
            "ReasonCodeId": "",
            "Hours": 1.0,
            "HourType": "Standard time",
            "BillingCode": "ST01"
            "Premiums": []
        }]
    }],
    "DailyPlanCostItemEquipment": [{
        "ResourceId": "100019",
        "ApprovedStandardHours": 1.0,
        "PlannedStandardHours": "",
        "SubmittedStandardHours": 1.0,
        "EquipmentId": 8412,
        "EquipmentType": "20-26",
        "ReasonCodes": [{
            "OperatedEmployeeId": "",
            "ReasonCodeId": "OPT",
            "Hours": 1.0
            "BillingCode": "ST01"
        }]
    }],
    "DailyPlanCostItemComponent": [],
    "DailyPlanCostItemVendors": [],
    "CostItemSourceSystemId": "efdc2f94951848a085b9eacfa0eba252",
    "WBSCode": "1010"
}, {
    "CostItemId": "1366683",
    "TaskPriority": 3,
    "PlanQuantity": 0.0,
    "SubmittedQuantity": 0.25,
    "ApprovedQuantity": 0.25,
    "DailyPlanCostItemEmployee": [{
        "ResourceId": "00493147",
        "SubmittedDoubletimeHours": 0.0,
        "SubmittedOvertimeHours": 0.0,

```

```

"SubmittedStandardHours": 8.0,
"ApprovedDoubletimeHours": 0.0,
"ApprovedOvertimeHours": 0.0,
"ApprovedStandardHours": 8.0,
"PlannedDoubletimeHours": "",
"PlannedOvertimeHours": "",
"PlannedStandardHours": "",
"EmployeeId": 186740,
"CraftCode": "OPJA",
"BillingClass": "",
"ReasonCodes": [{
  "ReasonCodeId": "",
  "Hours": 8.0,
  "HourType": "Standard time",
  "BillingCode": "ST01"
  "Premiums": []
}]
}, {
"ResourceId": "00493233",
"SubmittedDoubletimeHours": 0.0,
"SubmittedOvertimeHours": 0.0,
"SubmittedStandardHours": 8.0,
"ApprovedDoubletimeHours": 0.0,
"ApprovedOvertimeHours": 0.0,
"ApprovedStandardHours": 8.0,
"PlannedDoubletimeHours": "",
"PlannedOvertimeHours": "",
"PlannedStandardHours": "",
"EmployeeId": 186846,
"CraftCode": "ZWPJ",
"BillingClass": "",
"ReasonCodes": [{
  "ReasonCodeId": "",
  "Hours": 8.0,
  "HourType": "Standard time",
  "BillingCode": "ST01"
  "Premiums": []
}]
}, {
"ResourceId": "00493146",
"SubmittedDoubletimeHours": 0.0,
"SubmittedOvertimeHours": 0.0,
"SubmittedStandardHours": 8.0,
"ApprovedDoubletimeHours": 0.0,
"ApprovedOvertimeHours": 0.0,
"ApprovedStandardHours": 8.0,
"PlannedDoubletimeHours": "",
"PlannedOvertimeHours": "",
"PlannedStandardHours": "",
"EmployeeId": 186739,
"CraftCode": "OPJA",
"BillingClass": "",
"ReasonCodes": [{
  "ReasonCodeId": "",
  "Hours": 8.0,
  "HourType": "Standard time",
  "BillingCode": "ST01"

```

```

        "Premiums": []
    }
  ],
  "DailyPlanCostItemEquipment": [
    {
      "ResourceId": "100019",
      "ApprovedStandardHours": 5.0,
      "PlannedStandardHours": "",
      "SubmittedStandardHours": 5.0,
      "EquipmentId": 8412,
      "EquipmentType": "20-26",
      "ReasonCodes": [
        {
          "OperatedEmployeeId": "",
          "ReasonCodeId": "OPT",
          "Hours": 5.0,
          "BillingCode": "ST01"
        }
      ]
    }
  ],
  "DailyPlanCostItemComponent": [],
  "DailyPlanCostItemVendors": [],
  "CostItemSourceSystemId": "9f52ad940f7d4b899b619704fdc78946",
  "WBSCode": "1013"
},
"DailyPlanMaintenance": [],
"DailyPlanNote": [
  {
    "NoteId": 6773466,
    "Description": "xxxxxxxxx -",
    "DailyPlanNoteEmployee": [
      {
        "EmployeeDisplay": "00493147"
      },
      {
        "EmployeeDisplay": "00493146"
      }
    ],
    "DailyPlanNoteEquipment": [],
    "DailyPlanNoteTag": [
      {
        "DailyPlanNoteTagId": 4198901,
        "TagCode": "007"
      }
    ],
    "DailyPlanNoteTask": [],
    "DailyPlanNoteMaintenance": [],
    "DailyPlanNoteVendors": []
  },
  {
    "NoteId": 6773467,
    "Description": "Signed out on behalf of Executor",
    "DailyPlanNoteEmployee": [
      {
        "EmployeeDisplay": "00493233"
      },
      {
        "EmployeeDisplay": "00493147"
      },
      {
        "EmployeeDisplay": "00493146"
      }
    ],
    "DailyPlanNoteEquipment": [],
    "DailyPlanNoteTag": [],
    "DailyPlanNoteTask": [],
    "DailyPlanNoteMaintenance": [],
    "DailyPlanNoteVendors": []
  }
],
"DailyPlanSignoff": [],
"DailyPlanSignIns": [],

```

```

"DailyPlanBreaks": [{
  "EmployeeDisplay": "00493147",
  "BreakNumber": 1,
  "BreakStartTime": "2022/06/23 14:16:08",
  "BreakInMinutes": 5
}, {
  "EmployeeDisplay": "00493146",
  "BreakNumber": 3,
  "BreakStartTime": "2022/08/12 15:41:52",
  "BreakInMinutes": 5
}],
"DailyPlanBreakDetail": {
  "DailyPlanLevelBreaks": [{
    "BreakNumber": 1,
    "BreakStartTime": "2022/08/12 08:40:28",
    "BreakInMinutes": 10
  }, {
    "BreakNumber": 2,
    "BreakStartTime": "2022/08/12 11:40:36",
    "BreakInMinutes": 30
  }],
  "EmployeeLevelBreaks": [{
    "EmployeeDisplay": "00493147",
    "BreakNumber": 1,
    "BreakStartTime": "2022/06/23 14:16:08",
    "BreakInMinutes": 5
  }, {
    "EmployeeDisplay": "00493146",
    "BreakNumber": 3,
    "BreakStartTime": "2022/08/12 15:41:52",
    "BreakInMinutes": 5
  }
]
},
"Executers": [{
  "DisplayId": "",
  "RoleName": "Foreman",
  "ExecutorPosition": 1
}],
"Approvers": [{
  "DisplayId": "",
  "RoleName": "Superintendent",
  "ApproverPosition": 1
}],
"DailyPlanEmployeePayrollIndicator": [{
  "DailyPlanEmployeePayrollIndicatorId": 1069968,
  "EmployeeId": "186739",
  "EmployeeDisplay": "00493146",
  "PayrollIndicatorId": 72,
  "DailyPlanNoteId": 6773466,
  "CostItemId": "",
  "PayrollIdentifier": "007",
  "WBSPhaseCode": "",
  "EmployeeSourceSystemId": "00493146"
}, {
  "DailyPlanEmployeePayrollIndicatorId": 1069969,
  "EmployeeId": "186740",
  "EmployeeDisplay": "00493147",

```

```

        "PayrollIndicatorId": 72,
        "DailyPlanNoteId": 6773466,
        "CostItemId": "",
        "PayrollIdentifier": "007",
        "WBSPhaseCode": "",
        "EmployeeSourceSystemId": "00493147"
    }],
    "DailyPlanClientSignOff": [],
    "ShiftDetails": {
        "Shift": "01",
        "ShiftStartDateTime": "2022/08/12 08:00:00",
        "ShiftEndDateTime": "2022/08/12 17:00:00",
        "EmployeeShiftDetails": []
    },
    "ClientSignoffReportRecipients": []
}]]

```

## Example 2: Daily Plan with components and hours worked in Overtime, Double time, Standard time

```

[ {
    "PlanId": 95052,
    "CreatedDate": "2022/08/11 21:32:24",
    "ProjectId": "103755",
    "LanguageKey": "EN",
    "Location": "",
    "CreatedById": "",
    "DeviceKey": "",
    "Shift": "First Shift",
    "PlanDate": "2022/08/11 00:00:00",
    "PlanTitle": "Sample payload",
    "PlanStatusCode": "APPR",
    "ModifiedById": "",
    "ModifiedDate": "2022/08/11 21:39:15",
    "PlannedDate": "2022/08/11 21:36:24",
    "PlannedById": "",
    "ExecutedDate": "2022/08/11 21:38:06",
    "ExecutedById": "",
    "ApprovedDate": "2022/08/11 21:39:15",
    "ApprovedById": "",
    "ErrorHandlingToken": "",
    "ExternalSyncStatus": 2,
    "IsMailSent": false,
    "DailyPlanCostItem": [ {
        "CostItemId": "87761",
        "TaskPriority": 1,
        "PlanQuantity": 0.0,
        "SubmittedQuantity": 0.0,
        "ApprovedQuantity": 0.0,
        "DailyPlanCostItemEmployee": [ {
            "ResourceId": "00950748",
            "SubmittedDoubletimeHours": "",
            "SubmittedOvertimeHours": "",
            "SubmittedStandardHours": 2.0,
            "ApprovedDoubletimeHours": "",
            "ApprovedOvertimeHours": ""
        }
    ]
    }
    ]
} ]

```

```

    "ApprovedStandardHours": 2.0,
    "PlannedDoubletimeHours": "",
    "PlannedOvertimeHours": "",
    "PlannedStandardHours": 2.0,
    "EmployeeId": 71924,
    "CraftCode": "SPC2",
    "BillingClass": "",
    "ReasonCodes": []
  }, {
    "ResourceId": "00961464",
    "SubmittedDoubletimeHours": "",
    "SubmittedOvertimeHours": "",
    "SubmittedStandardHours": 2.0,
    "ApprovedDoubletimeHours": "",
    "ApprovedOvertimeHours": "",
    "ApprovedStandardHours": 2.0,
    "PlannedDoubletimeHours": "",
    "PlannedOvertimeHours": "",
    "PlannedStandardHours": 2.0,
    "EmployeeId": 70358,
    "CraftCode": "STST",
    "BillingClass": "",
    "ReasonCodes": []
  }, {
    "ResourceId": "00902351",
    "SubmittedDoubletimeHours": "",
    "SubmittedOvertimeHours": "",
    "SubmittedStandardHours": 2.0,
    "ApprovedDoubletimeHours": "",
    "ApprovedOvertimeHours": "",
    "ApprovedStandardHours": 2.0,
    "PlannedDoubletimeHours": "",
    "PlannedOvertimeHours": "",
    "PlannedStandardHours": 2.0,
    "EmployeeId": 26508,
    "CraftCode": "SPGF",
    "BillingClass": "",
    "ReasonCodes": []
  }],
  "DailyPlanCostItemEquipment": [{
    "ResourceId": "123993",
    "ApprovedStandardHours": 8.0,
    "PlannedStandardHours": 8.0,
    "SubmittedStandardHours": 8.0,
    "EquipmentId": 1737,
    "EquipmentType": "02-08",
    "ReasonCodes": [{
      "OperatedEmployeeId": "",
      "ReasonCodeId": "OPT",
      "Hours": 8.0
    }],
    "BillingCode": "ST01"
  }],
  "DailyPlanCostItemComponent": [],
  "DailyPlanCostItemVendors": [],
  "CostItemSourceSystemId": "3e7374dfdb9e44eb960df96b2f6d3c37",
  "WBSCode": "1011"

```

```

    }, {
      "CostItemId": "87774",
      "TaskPriority": 2,
      "PlanQuantity": 0.0,
      "SubmittedQuantity": 0.0,
      "ApprovedQuantity": 0.0,
      "DailyPlanCostItemEmployee": [{
        "ResourceId": "00961464",
        "SubmittedDoubletimeHours": "",
        "SubmittedOvertimeHours": "",
        "SubmittedStandardHours": 2.0,
        "ApprovedDoubletimeHours": "",
        "ApprovedOvertimeHours": "",
        "ApprovedStandardHours": 2.0,
        "PlannedDoubletimeHours": "",
        "PlannedOvertimeHours": "",
        "PlannedStandardHours": 2.0,
        "EmployeeId": 70358,
        "CraftCode": "STST",
        "BillingClass": "",
        "ReasonCodes": []
      }],
    }, {
      "ResourceId": "00950748",
      "SubmittedDoubletimeHours": "",
      "SubmittedOvertimeHours": "",
      "SubmittedStandardHours": 2.0,
      "ApprovedDoubletimeHours": "",
      "ApprovedOvertimeHours": "",
      "ApprovedStandardHours": 2.0,
      "PlannedDoubletimeHours": "",
      "PlannedOvertimeHours": "",
      "PlannedStandardHours": 2.0,
      "EmployeeId": 71924,
      "CraftCode": "SPC2",
      "BillingClass": "",
      "ReasonCodes": []
    }, {
      "ResourceId": "00902351",
      "SubmittedDoubletimeHours": "",
      "SubmittedOvertimeHours": "",
      "SubmittedStandardHours": 2.0,
      "ApprovedDoubletimeHours": "",
      "ApprovedOvertimeHours": "",
      "ApprovedStandardHours": 2.0,
      "PlannedDoubletimeHours": "",
      "PlannedOvertimeHours": "",
      "PlannedStandardHours": 2.0,
      "EmployeeId": 26508,
      "CraftCode": "SPGF",
      "BillingClass": "",
      "ReasonCodes": []
    }],
    "DailyPlanCostItemEquipment": [{
      "ResourceId": "124635",
      "ApprovedStandardHours": 8.0,
      "PlannedStandardHours": 8.0,
      "SubmittedStandardHours": 8.0,
    }],
  }

```

```

    "EquipmentId": 3045,
    "EquipmentType": "02-08",
    "ReasonCodes": [{
      "OperatedEmployeeId": "",
      "ReasonCodeId": "OPT",
      "Hours": 8.0
      "BillingCode": "ST01"
    }]
  }],
  "DailyPlanCostItemComponent": [],
  "DailyPlanCostItemVendors": [],
  "CostItemSourceSystemId": "d1133727d101496fb98736633f0b15b8",
  "WBSCode": "1026"
}, {
  "CostItemId": "87892",
  "TaskPriority": 3,
  "PlanQuantity": 5408.0,
  "SubmittedQuantity": 0.0,
  "ApprovedQuantity": 0.0,
  "DailyPlanCostItemEmployee": [{
    "ResourceId": "00902351",
    "SubmittedDoubletimeHours": "",
    "SubmittedOvertimeHours": "",
    "SubmittedStandardHours": 4.0,
    "ApprovedDoubletimeHours": "",
    "ApprovedOvertimeHours": "",
    "ApprovedStandardHours": 4.0,
    "PlannedDoubletimeHours": "",
    "PlannedOvertimeHours": "",
    "PlannedStandardHours": 4.0,
    "EmployeeId": 26508,
    "CraftCode": "SPGF",
    "BillingClass": "",
    "ReasonCodes": []
  }, {
    "ResourceId": "00950748",
    "SubmittedDoubletimeHours": "",
    "SubmittedOvertimeHours": "",
    "SubmittedStandardHours": 4.0,
    "ApprovedDoubletimeHours": "",
    "ApprovedOvertimeHours": "",
    "ApprovedStandardHours": 4.0,
    "PlannedDoubletimeHours": "",
    "PlannedOvertimeHours": "",
    "PlannedStandardHours": 4.0,
    "EmployeeId": 71924,
    "CraftCode": "SPC2",
    "BillingClass": "",
    "ReasonCodes": []
  }, {
    "ResourceId": "00961464",
    "SubmittedDoubletimeHours": "",
    "SubmittedOvertimeHours": "",
    "SubmittedStandardHours": 4.0,
    "ApprovedDoubletimeHours": "",
    "ApprovedOvertimeHours": "",
    "ApprovedStandardHours": 4.0,

```



```

        "PlannedDoubletimeHours": "",
        "PlannedOvertimeHours": "",
        "PlannedStandardHours": 4.0,
        "EmployeeId": 70358,
        "CraftCode": "STST",
        "BillingClass": "",
        "ReasonCodes": []
    }],
    "DailyPlanCostItemEquipment": [{
        "ResourceId": "124636",
        "ApprovedStandardHours": 8.0,
        "PlannedStandardHours": 8.0,
        "SubmittedStandardHours": 8.0,
        "EquipmentId": 2026,
        "EquipmentType": "02-08",
        "ReasonCodes": [{
            "OperatedEmployeeId": "",
            "ReasonCodeId": "OPT",
            "Hours": 8.0
        }],
        "BillingCode": "ST01"
    }],
    "DailyPlanCostItemComponent": [],
    "DailyPlanCostItemVendors": [],
    "CostItemSourceSystemId": "a5ff4f7de5294607b5efc80196a51360",
    "WBSCode": "1246"
}],
"DailyPlanMaintenance": [],
"DailyPlanNote": [{
    "NoteId": 78524,
    "Description": "Test note",
    "DailyPlanNoteEmployee": [{
        "EmployeeDisplay": "00902351"
    }],
    "DailyPlanNoteEquipment": [],
    "DailyPlanNoteTag": [{
        "DailyPlanNoteTagId": 4342,
        "TagCode": "13"
    }],
    "DailyPlanNoteTask": [{
        "CostItemId": 87761
    }],
    "DailyPlanNoteMaintenance": [],
    "DailyPlanNoteVendors": []
}],
"DailyPlanSignoff": [],
"DailyPlanSignIns": [],
"DailyPlanBreaks": [{
    "EmployeeDisplay": "00961464",
    "BreakNumber": 3,
    "BreakStartTime": "2022/08/11 15:30:00",
    "BreakInMinutes": 5
}],
"DailyPlanBreakDetail": {
    "DailyPlanLevelBreaks": [{
        "BreakNumber": 1,
        "BreakStartTime": "2022/08/11 11:00:00",
    }],
}

```

```

        "BreakInMinutes": 20
    }, {
        "BreakNumber": 2,
        "BreakStartTime": "2022/08/11 13:00:00",
        "BreakInMinutes": 25
    }],
    "EmployeeLevelBreaks": [{
        "EmployeeDisplay": "00961464",
        "BreakNumber": 3,
        "BreakStartTime": "2022/08/11 15:30:00",
        "BreakInMinutes": 5
    }
]
},
"Executers": [{
    "DisplayId": "00323170",
    "RoleName": "Progress Foreman",
    "ExecutorPosition": 1
}],
"Approvers": [{
    "DisplayId": "",
    "RoleName": "Superintendent",
    "ApproverPosition": 1
}],
"DailyPlanEmployeePayrollIndicator": [],
"DailyPlanClientSignOff": [{
    "Name": "Luke",
    "Email": ""
}],
"ShiftDetails": {
    "Shift": "First Shift",
    "ShiftStartDateTime": "2022/08/11 06:40:00",
    "ShiftEndDateTime": "2022/08/12 16:00:00",
    "EmployeeShiftDetails": []
},
"ClientSignoffReportRecipients": ["luke@email.com"]
}]

```

## Daily Plan Status

From the external system, API DailyPlanStatus\_Import is responsible for updating daily plan status to Final Processing Complete.

<b>Direction</b>	To the InEight cloud platform	
<b>Frequency</b>	Determined by external system	
<b>Trigger Methods</b>	Determined by external system	
<b>Average Payload Size</b>	Determined by external system. Messages can be sent containing a single daily plan update as they are linearly processed by an external system, or the message can contain all daily plan updates for a day or week if the external processing happens in batches.	
<b>API Name</b>	DailyPlanStatus_Import	
<b>InEight Application</b>	<b>Starting Version</b>	18.1
	<b>Ending Version</b>	

## Fields

Depth	Name	Type	Precision	Parent	Req.
1	DailyPlanId	Integer	Int64		Yes
1	PlanStatusCode	String	100		Yes
1	ModifiedById	String	50		No
1	ModifiedDate <sup>2</sup>	String	25		No
1	ReleasedDate <sup>2</sup>	String	25		No
1	ReleasedById	String	50		No
1	Errors	List			No

## Field Descriptions

Name	Description	Example
DailyPlanId	Unique identifier of the daily plan.	2134
PlanStatusCode	Sets the incoming status of the daily plan. Possible values that can be sent are: <ul style="list-style-type: none"> <li>FINL - Updates the status of the daily plan to Final Processing Complete and sets the Payroll Sync attribute of the daily plan to Success.</li> <li>FAIL - Does not change the existing status of the daily plan but sets the Payroll Sync attribute of the daily plan to Fail</li> </ul>	FINL

Name	Description	Example
ModifiedById	DisplayUserId of the user that last modified the daily plan.	90802983
ModifiedDate	Date on which the plan was modified in the external system.	2018-02-21 20:26:35
ReleasedDate	Date on which the daily plan was released by the external system.	2018-02-21 20:26:35
ReleasedById	DisplayUserId of the user that released the daily plan.	90802983
Errors	Optional list of messages that can be provided when Plan Status Code is set to FAIL. Messages consist of text characters and the length is unlimited. Each message should be separated by a comma in the array list.	Invalid Employee 908029, Invalid Phase Code, Error on processing server

## Error Messages

The following error messages are generated by the InEight cloud platform and products 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 in the InEight cloud platform and products to 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 logging.

API/Entity Logic	Condition	Code	Message
API Validation	All received records have been validated.	200	
API Validation	An exception occurred but was handled by the integration.	500	Exception occurred while processing in PlanDailyPlanPayrollConfirmation
API Validation	Payload in the body of the message was empty.	200	

## Sample JSON

```
[
  {
    "DailyPlanId": "1560",
    "PlanStatusCode": "FINL",
    "ModifiedById": "00206457",
    "ModifiedDate": "2018-02-21 20:26:35",
    "ReleasedDate": "2018-02-21 20:26:35",
    "ReleasedById": "00206457",
    "Errors": [
      "Error Message 1",
      "Error Message 2"
    ]
  }
]
```

```

    ],
    {
      "DailyPlanId": "1561",
      "PlanStatusCode": "FAIL",
      "ModifiedById": "00206457",
      "ModifiedDate": "2018-02-21 20:26:35",
      "ReleasedDate": "2018-02-21 20:26:35",
      "ReleasedById": "00206457",
      "Errors": []
    }
  ]

```

## Verification

The InEight Progress UI on the Daily Plans page shows a list of daily plans and their status. In the following example, it shows updated status of Final Processing Complete where the Payroll Sync status was updated to Success and no status change for those that show Fail. Additionally, the ModifiedById and ModifiedDate JSON fields appear as columns in the UI, as **Updated by** and **Last updated on**, respectively.

**NOTE:** Messages sent in the Errors array list are currently not available in the UI.

Daily Plans		MY DAILY PLANS				ALL PLANS			
Plan ID	Plan date	Plan name	Shift	Location	Work package ID	Status	Payroll Sync		
23447	Thu, 16 Aug 2018	TC_232070	FIRST SHIFT			Final Processing Complete	Success		
23461	Thu, 16 Aug 2018	signout	TC_450054210			Final Processing Complete	Success		
23465	Thu, 16 Aug 2018	client sign	TC_450054210			Final Processing Complete	Success		
23468	Thu, 16 Aug 2018	TC_290967	FIRST SHIFT			Final Processing Complete	Success		
23375	Wed, 15 Aug 2018	Approving hours only...	FIRST SHIFT			Final Processing Complete	Success		
23037	Mon, 13 Aug 2018	20180813191729	FIRST SHIFT			In Planning	Fail		
23055	Mon, 13 Aug 2018	20180813203533	FIRST SHIFT			Execution	Fail		
23057	Mon, 13 Aug 2018	20180813203533	FIRST SHIFT			In Planning	Fail		
22465	Fri, 10 Aug 2018	20180810222702	FIRST SHIFT			Execution	Fail		
22467	Fri, 10 Aug 2018	20180810222702	FIRST SHIFT			In Planning	Fail		

## Employee Work Schedule

This integration allows customers to provide a simple listing of the scheduled days that employees should be at work and for the number of hours they are scheduled. This information is used for lookup on validations that employee data entered in a daily plan (employee, date, hours) falls within expected parameters of their planned work schedule.

Data provided in this integration is retained by the InEight cloud platform for 14 days. When the integration is performed and records are received, a combination of EmployeeId and WorkDate is used by the InEight cloud platform to determine if an existing record should be updated or a new record created.

A record should be provided for every date that an employee has a known work schedule. On days when the employee is expected to work, a non-zero value should be provided as the scheduled hours. On days when an employee is scheduled to be off work, a zero should be provided as the scheduled hours. The only time a record should not be provided is when the schedule for the employee is unknown.

<b>Direction</b>	To the InEight cloud platform.	
<b>Frequency</b>	Determined by external system.	
<b>Trigger Methods</b>	Determined by external system.	
<b>Average Payload Size</b>	Entire entity. Because this integration is for all employees across all projects, there are potentially thousands, or tens of thousands of records possible in each integration request.	
<b>APIM Name</b>	EmployeeWorkSchedule_Import	
<b>InEight Application</b>	<b>Starting Version</b>	18.3
	<b>Ending Version</b>	

### Fields

Depth	Name	Type	Precision <sup>1</sup>	Parent	Req.
1	EmployeeId <sup>2</sup>	String	50		Yes
1	WorkDate <sup>2</sup>	String	25		Yes
1	ScheduledHours	Decimal	16,5		Yes

*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
EmployeeId	ID of the employee for the record. A combination of EmployeeId and WorkDate is considered a unique record, which makes it possible for an employee to have multiple work dates sent in a single integration request. EmployeeId in this integration must match a valid SourceSystemId for an employee record in the InEight cloud platform.	1346735
WorkDate	UTC formatted date/time string. Only the date is used, the time component of the string is ignored.	2018-05-24T00:00:00+00:00
ScheduledHours	Number of hours the employee is expected to work on the WorkDate. For a scheduled day off, send a value of 0. The value sent in this field must be in quarter-hour increments with two decimal places.	8.25

## Error Messages

The following error messages are generated by the InEight cloud platform and products 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 in the InEight cloud platform and products to 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 logging.

API/Entity Logic	Condition	Code	Message Body
API Validation	All received records have been validated.	200	
API Validation	WorkDate is not present in a record. Each failed record will have a separate message in the response body.	200	Schedule record for {EmployeeId} contains either a blank or an invalid value for work date and the record is ignored. Please fix the error and re-import.
API Validation	ScheduleHours contains an invalid value in a record. Each failed record will have a separate message in the response body.	200	Schedule record for {EmployeeId} contains an invalid value for Scheduled hours and the record is ignored. Please fix the error and re-import.
API Validation	EmployeeId is not present in a record. Each failed record will have a separate message in the response body.	200	Schedule record contains a blank employee ID when employee ID is a mandatory field. The record is ignored. Please fix the error and re-import.

## Sample JSON

```
[
  {
    "EmployeeId": "12345",
    "WorkDate": "2015-09-25T00:00:00+00:00",
    "ScheduledHours": 2
  },
  {
    "EmployeeId": "12345",
    "WorkDate": "2015-09-26T00:00:00+00:00",
    "ScheduledHours": 7.25
  },
  {
    "EmployeeId": "12345",
    "WorkDate": "2015-09-27T00:00:00+00:00",
    "ScheduledHours": 7.25
  },
  {
    "EmployeeId": "12346",
    "WorkDate": "2015-09-25T00:00:00+00:00",
    "ScheduledHours": 8
  },
  {
    "EmployeeId": "12346",
    "WorkDate": "2015-09-26T00:00:00+00:00",
    "ScheduledHours": 0
  },
  {
    "EmployeeId": "12346",
    "WorkDate": "2015-09-27T00:00:00+00:00",
    "ScheduledHours": 8
  }
]
```

## Verification

There is nothing available in the InEight UI to verify the data received, and at this time data is used internally to perform specific payroll validations in InEight Plan.



## Daily Plan Work Orders (Post)

<b>Direction</b>	To the InEight cloud platform.	
<b>Frequency</b>	Updated as new work orders are created in the work order system of record, or batch as needed.	
<b>Trigger Methods</b>	Determined by external system.	
<b>Average Payload Size</b>	Tens of records daily or weekly as new work orders are added from the external system. Updates can be on a record-by-record basis depending on how the external system triggers updates.	
<b>APIM Name</b>	DailyPlanWorkOrders_Import	
<b>InEight Application</b>	<b>Starting Version</b>	18.2
	<b>Ending Version</b>	

### Fields

Depth	Name	Type	Precision <sup>1</sup>	Parent	Req.
1	WorkOrderDisplayId	String	50		Yes
1	EquipmentSourceSystemId <sup>2</sup>	String	50		Yes
1	EquipmentId	NA	NA		No
1	WorkOrderStatusDisplay	String	50		Yes
1	Description	String	100		Yes
1	SourceSystemId <sup>2</sup>	String	100		Yes
1	SourceSystemName	String	100		Yes
1	WorkOrderTypeDisplay	String	50		Yes
1	ProjectDisplay	String	200		Yes
1	ProjectId	NA	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
WorkOrderDisplayId	Unique identifier of the work order created by the system of record that is understood by users in InEight cloud platform when searching for work orders.	2222
EquipmentSourceSystemId	Unique identifier of the equipment that the work order is for. This must match an existing record from the Equipment integration.	159116
EquipmentId	<i>Do not use. This field will be deprecated in future versions of this integration.</i>	
WorkOrderStatusDisplay	Status to be displayed for the work order in InEight cloud platform. Possible values that can be sent for this field are Open and Closed	Closed
Description	Text name and description of the work order to help users when searching for work orders.	Repair Crane 19800
SourceSystemId	Unique identifier for the work order that is created by the system of record and used as the method of specifying records when exchanging data between systems.	14911656
SourceSystemName	Name of the source system from where the data is sourced.	JDE
WorkOrderTypeDisplay	Text value representing the type of work order that is shown when users search for work orders in the InEight cloud platform. The values allowed for this field are preloaded into the InEight cloud platform and cannot be edited in current versions. Possible values are: <ul style="list-style-type: none"> <li>• 500, Preventative Maintenance</li> <li>• 510, Engine &amp; Exhaust</li> <li>• 520, Power Train</li> <li>• 530, Frame &amp; Cab</li> <li>• 540, Undercarriage or Tires</li> <li>• 550, Implements</li> <li>• 560, Electrical System</li> <li>• 570, Hydraulic System</li> <li>• 580, Cooling System</li> <li>• 590, Brakes or Steering System</li> <li>• 600, Air or Lube System</li> <li>• 610, Standing Work Order</li> <li>• NA, Not Applicable</li> </ul>	500
ProjectDisplay	Unique identifier of the project that the work order is assigned.	1101111
ProjectId	<i>Do not use. This field will be deprecated in future versions of this integration.</i>	

## Error Messages

The following error messages are generated by the InEight cloud platform and products 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 in the InEight cloud platform and products to 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 logging.

API/Entity Logic	Condition	Code	Message
API Validation	DailyPlanWorkOrder List is null or empty	400	BadRequest
API Validation	Mandatory fields are null or empty Error Fields: <ul style="list-style-type: none"> <li>• Description</li> <li>• WorkOrderDisplayId</li> <li>• WorkOrderStatusDisplay</li> <li>• SourceSystemName</li> <li>• WorkOrderTypeDisplay</li> <li>• ProjectDisplay</li> <li>• EquipmentSourceSystemId</li> </ul>	200	ErrorMessage: Mandatory field(s) null or empty
API Validation	Mandatory fields are invalid Error Fields: <ul style="list-style-type: none"> <li>• ProjectDisplay</li> <li>• EquipmentSourceSystemId</li> </ul>	200	ErrorMessage: Mandatory field(s) value invalid
API Validation	Exception happened in Plan Service	200	Insert failed

## Sample JSON

```
[
  {
    "WorkOrderDisplayId": "2222",
    "EquipmentSourceSystemId": "159116 SSID",
    "WorkOrderStatusDisplay": "Closed",
    "Description": "Repair Crane 19800",
    "SourceSystemId": "14911656 SSID",
    "SourceSystemName": "14911656 SSID",
    "WorkOrderTypeDisplay": "500",
    "ProjectDisplay": "110111"
  }, {
    "WorkOrderDisplayId": "1111",
    "EquipmentSourceSystemId": "163017 SSID",
    "WorkOrderStatusDisplay": "Open",
    "Description": "Repair Caterpillar 000356",
    "SourceSystemId": "123017 SSID",
    "SourceSystemName": "SSID",
    "WorkOrderTypeDisplay": "590",
    "ProjectDisplay": "110111"
  }
]
```

```

}, {
  "WorkOrderDisplayId": "3333",
  "EquipmentSourceSystemId": "172101 SSID",
  "WorkOrderStatusDisplay": "Open",
  "Description": "Repair Caterpillar 000675",
  "SourceSystemId": "1821012 SSID",
  "SourceSystemName": "SSID",
  "WorkOrderTypeDisplay": "530",
  "ProjectDisplay": "103361"
}
]

```

## Verification

Work orders with a status of Open can be selected from the drop-down list in a daily plan time sheet and assigned to a cost item/hours. Work orders that have a status of Closed, will not be shown in the drop-down list.

## Quantity Claiming Detail (Get)

This integration allows customers to retrieve all individual claiming transactions that have been approved from daily plans or quantity tracking within InEight Plan and InEight Progress.

To externalize the quantities claimed for components associated with the WBS, the claims will be sent to both existing internal integration sources and new external integration systems. The external source will be able to see the new claim from Plan by using the pull pattern in the APIM external interface. One request will serve the maximum of 50 claims. This external data API will be controlled (enabled or disabled) through a tenant configuration PublishQCD on demand from the customers.

When quantity claims occur in quantity tracking or a daily plan with quantities reaches an approval status, these transactional records are made available. For customers using InEight Control, claiming details are provided to Control where their associated cost item can be updated further through indirect claims, such as overhead and rental fees, before being sent to an ERP system.

**NOTE:**

This integration requires a specific configuration in the Tenant Catalog for the account to make the data available. Customers should consult with their account representative or the InEight DevOps team to add the PublishQCD catalog setting or to update the value to true.

For customers that are not using InEight Control, this integration provides a method to capture claiming details against a cost item directly from InEight Plan and InEight Progress as they become available, and then feed them into an ERP or other accounting system.

- Whenever the integration API is called, it returns the available data in batches of up to 50 records at a time. Batch sizes can be smaller based on real-time system performance to avoid time-outs and system errors.
- The calling system should recursively call this integration API until no records are received in the API, which indicates there is no more data available.
- After the API returns the data, the data is then removed from the queue and is no longer available through the integration API. It is assumed that the calling system owns the data after it is received through this API response and it can manage any issues while appropriately processing the data, as necessary.
- Unlike with the Daily Plan integration, there is no confirmation/acknowledgement for transactions as part of this API.
- The quantities values in each record represent the amount that was claimed in the transaction and are not rolled up or current totals for the components on which the transaction occurred. It is the responsibility of the receiving system to keep track of totals.

<b>Direction</b>	From the InEight cloud platform.	
<b>Frequency</b>	Data can be retrieved from InEight cloud platform at any time. Records containing claiming information are made available to be retrieved after claiming has been placed into an approved status in InEight Plan.	
<b>Trigger Methods</b>	Determined by external system.	
<b>Average Payload Size</b>	Dependent on project size. Each daily plan or direct claiming in Quantity Tracking can contain tens of records. There are potentially many daily plans and direct claiming occurrences per day in a project.	
<b>APIM Name</b>	QuantityClaimDetail_Get	
<b>InEight Application</b>	<b>Starting Version</b>	18.3
	<b>Ending Version</b>	

## Fields

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	QuantityClaimDetailId	Number	Integer	
1	QuantityClaimId	Number	Integer	
1	GroupId	Number	Integer	
1	Executor	String	50	
1	Approver	String	50	
1	Notes	String	4000	
1	ClaimedDate <sup>2</sup>	String	25	
1	EarnedQuantity	Decimal	28,15	
1	ComponentToDateQuantity	Decimal	28,15	
1	ClaimedQuantity	Decimal	28,15	
1	CostItemId	Number	Integer	
1	CreatedBy	String	50	
1	CreatedDate <sup>2</sup>	String	25	
1	DailyPlanId	Number	Integer	
1	ProjectDisplay	String	200	
1	WBSPhaseCode	String	50	
1	CostItemSourceSystemId	String	50	

*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 - 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
QuantityClaimDetailId	Unique ID generated in InEight Plan for each quantity claim transaction.	263963
QuantityClaimId	Internal ID association of claiming scheme step and component.	211739
GroupId	Internal sequence ID generated on each claim against a WBS.	191154
Executor	SourceSystemId from the user record for the foreman or the executor of the daily plan.	joedoe@somewhere.com
Approver	SourceSystemId from the user record for the person who approved the daily plan or for the person claiming it in quantity tracking.	joedoe@somewhere.com
Notes	Free form field that is used for additional information against this claim transaction.	A note goes here
ClaimedDate	Date when the work was done as reported by the person claiming it.	2019-04-09T00:00:00+00:00
EarnedQuantity	Effective claim that indicates the progress of work done against the WBS Phase Code.	1
ComponentToDateQuantity	Total claimed units of quantity at the component level.	1
ClaimedQuantity	User entered value against a claiming step on a component, which might not be the effective roll up value against a cost item (WBS phase code). This can be used for audit/reporting purposes.	1
CostItemId	Unique ID used internally in the InEight cloud platform for the task/cost item.	495800
CreatedBy	SourceSystemId of the user who approve the quantities.	joedoe@somewhere.com
CreatedDate	Date when the record was created.	2019-04-09T18:55:41.8515103+00:00
DailyPlanId	Internal InEight ID for the daily plan where claiming occurred. If claiming was performed in quantity tracking, this field will be null.	186
ProjectDisplay	Public project ID where claiming occurred.	1101111
WBSPhaseCode	WBS phase code for the cost item.	1009

Name	Description	Example
CostItemSourceSystemId	SourceSystemId for the cost item that was sent when importing the cost item into InEight Control.	cc85269f1dfe40f1a418dce16bc6d4f3

## Sample JSON

```
[
  {
    "QuantityClaimDetailId": 264465,
    "QuantityClaimId": 453622,
    "GroupId": 191458,
    "Executor": "",
    "Approver": "",
    "Notes": "System generated this log as there was a change in Component Quantity and Step Quantity after claim was started",
    "ClaimedDate": "2019-04-16T00:00:00+00:00",
    "EarnedQuantity": 0,
    "ComponentToDateQuantity": 0.48,
    "ClaimedQuantity": 0,
    "CostItemId": 324,
    "CreatedBy": "johndoe@INEIGHT.COM",
    "CreatedDate": "2019-04-16T09:18:25.9240854+00:00",
    "DailyPlanId": "",
    "ProjectDisplay": "103361",
    "WBSPhaseCode": "461",
    "CostItemSourceSystemId": ""
  }, {
    "QuantityClaimDetailId": 264472,
    "QuantityClaimId": 453640,
    "GroupId": 191458,
    "Executor": "",
    "Approver": "",
    "Notes": "",
    "ClaimedDate": "2019-04-16T00:00:00+00:00",
    "EarnedQuantity": 0,
    "ComponentToDateQuantity": 0.48,
    "ClaimedQuantity": 0,
    "CostItemId": 324,
    "CreatedBy": "janedoe@INEIGHT.COM",
    "CreatedDate": "2019-04-16T09:18:25.9240854+00:00",
    "DailyPlanId": "",
    "ProjectDisplay": "103361",
    "WBSPhaseCode": "461",
    "CostItemSourceSystemId": ""
  }
]
```



## Component Details

Due to the potential for a large set of results from this request, a file with the results is created in our blob storage and a URL to the file is provided in the Location field of the response.

This pattern is used for fetching the entities which are very complex, and it will take time to assemble. The API exposes a GET entity with parameters. A user will request the entity through an end point, and then a background process will take over and perform the work necessary to assemble the requested payload asynchronously. The original request will return a 202 Accepted response with a Location header containing a GET status URL that the client can call periodically to check on the status of the request. When the background process has completed processing the request and has assembled the payload (preferably cached it in storage), the Get status request will then return a 200 OK response with the payload.

Due to the potential for a large set of results from this request, a file with the results is created within our blob storage and a URL to the file is provided in the Location field of the response.

<b>Direction</b>		From the InEight cloud platform.
<b>Frequency</b>		Data can be retrieved from InEight cloud platform at any time.
<b>Trigger Methods</b>		Determined by external system.
<b>Average Payload Size</b>		Hundreds or thousands of records.
<b>APIM Name</b>		ComponentsAndCharacteristics_Request
<b>InEight Application</b>	<b>Starting Version</b>	19.1
	<b>Ending Version</b>	

## Supported Filters

Data provided by the InEight cloud platform to external systems (outbound), can support selective fetching of data by applying filters in the API request.

Filter Name	Data Type	Description
projectDisplay (mandatory)	String	Filters the returned records to a specific project. Only one projectDisplay can be given in the filter. Example Format: "103114"
lastSyncDate	String	If provided, the returned records are from that date forward.

## Fields

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	ComponentName	String	200	

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	DisciplineId	Number	Integer	
1	DisciplineName	String	100	
1	CommodityId	Number	Integer	
1	CommodityName	String	100	
1	ComponentLevelId	Number	Integer	
1	ComponentLevelName	String	50	
1	ClaimingSchemeId	Number	Integer	
1	ClaimingSchemeName	String	100	
1	ProjectId	Number	Integer	
1	ProjectDisplay	String	200	
1	Description	String	200	
1	Comments	String	200	
1	ComponentQuantity	Decimal	28,15	
1	ToDateQuantity	Decimal	28,15	
1	UoMId	Number	Integer	
1	UOMName	String		
1	WBS	String	50	
1	ScheduleId	String	50	
1	AGUGId	Number	Integer	
1	ConstructionAreaId	Number	Integer	
1	ConstructionAreaName	String	50	
1	Phase	String	50	
1	SystemId	Number	Integer	
1	SystemName	String	50	
1	Line	String	100	
1	TagNumber	String	50	
1	Specification	String	50	
1	CodeClass	String	50	
1	MaterialCode	String	50	
1	Size	String	50	
1	SizeCode	String	50	

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	Thickness	String	50	
1	Weight	String	50	
1	ITP	String	50	
1	Supplier	String	50	
1	ShopFieldId	Number	Integer	
1	TurnoverId	Number	Integer	
1	TurnoverName	String	50	
1	TestPackageId	Number	Integer	
1	TurnoverComplete <sup>2</sup>	String	Date	
1	TestPackageComplete <sup>2</sup>	String	Date	
1	IsDefault	Boolean	NA	
1	CostItemId	Number	Integer	
1	PercentageComplete	Decimal	28,15	
1	Id	Number	Integer	
1	CreatedByDisplayId	String	50	
1	CreatedDate <sup>2</sup>	String	Date	
1	ModifiedByDisplayId	String	50	
1	ModifiedDate <sup>2</sup>	String	Date	
1	IsActive	Boolean	NA	
1	AccountId	Number	Integer	
1	SubSystemId	Number	Integer	
1	SubSystemName	String	50	
1	ConstructionCommodityId	Number	Integer	
1	ConstructionCommodityName	String	50	
1	ComponentCharacteristics	Array	NA	
2	Name	String	100	ComponentCharacteristics
2	Value	String	1024	ComponentCharacteristics

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 - 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
ComponentName	Name given to the component displayed in the UI.	Pipe Checksheet
DisciplineId	Internal ID of the discipline associated to the component.	1324
DisciplineName	Name of the discipline.	Building
CommodityId	Internal ID of the commodity associated to the component.	546755
CommodityName	Name of the commodity.	Cable
ComponentLevelId	Internal ID of the component level. This determines if the component is a discipline or commodity.	64565
ComponentLevelName	Name of the component level. The values allowed in this field are: <ul style="list-style-type: none"> <li>• Discipline</li> <li>• Commodity</li> </ul>	Discipline
ClaimingSchemeId	Reference ID for a claiming scheme.	64677
ClaimingSchemeName	Name of the claiming scheme.	Concrete
ProjectId	Internal ProjectId reference.	103467
ProjectDisplay	Unique name or ID of a project used in screens.	103467
Description	Public description of the component.	Checklist used for any pipe operations
Comments	Free text comments for the component.	Last updated op May 04
ComponentQuantity	Quantity used for measuring progress against work in the component.	1
ToDateQuantity	Total claimed units of quantity at the component level.	1
UoMId	Internal reference ID for a UOM.	22
UOMName		
WBS	WBSPhaseCode of the cost item associated with the component.	10.01.123.1234
ScheduleId	Free-form text component attributes.	Underground Layer 1
AGUGId	Determines if the work is above or underground. Values that can appear in this field are: <ul style="list-style-type: none"> <li>• Above</li> <li>• Under</li> </ul>	Above
ConstructionAreaId	Reference ID for construction area.	17
ConstructionAreaName	Name of the construction area.	West End
Phase	Free-form text component attributes.	1A

Name	Description	Example
SystemId	Reference ID for a system.	5
SystemName	Name of the system.	Heating & Cooling
Line	Free-form text component attributes.	Any text
TagNumber	A unique tag number associated to the component.	TM654314J14
Specification	Free-form text component attributes.	Any text
CodeClass	Free-form text component attributes.	Any text
MaterialCode	Free-form text component attributes.	Any text
Size	Free-form text component attributes.	Any text
SizeCode	Free-form text component attributes.	Any text
Thickness	Free-form text component attributes.	Any text
Weight	Free-form text component attributes.	Any text
ITP	Free-form text component attributes.	Any text
Supplier	Free-form text component attributes.	Any text
ShopFieldId	Internal ID of a shop field.	54
TurnoverId	Internal reference ID for a turnover package.	34
TurnoverName	Name of the turnover package.	Midtown Station
TestPackageId	component ID associated with claiming scheme Test Package.	654644
TurnoverComplete	Turn over complete date.	2019-05-21 05:00:00.0000000 +00:00
TestPackageComplete	Test package complete date	2019-05-21 05:00:00.0000000 +00:00
IsDefault	Boolean value identifier to check whether the component is default or not.	True/false
CostItemId	ID of a cost item associated to the component.	311331
PercentageComplete	Percentage of claim completed for the component.	100
Id	Internal component ID.	646744
CreatedByDisplayId	DisplayId for the user that created the component.	joedoe@somewhere.com
CreatedDate	Date on which the component was created.	2019-05-14 13:07:31.0929640 +00:00
ModifiedByDisplayId	DisplayId for the user that last modified the component.	joedoe@somewhere.com
ModifiedDate	Date on which the component was last modified.	2019-05-14 13:07:31.0929640 +00:00

Name	Description	Example
IsActive	Sending a value of false in this field causes the component record to be soft deleted from the InEight cloud platform. If a value is not provided, the default value true is used.	true
AccountId	Identifier of the customer account number in the InEight environment tenant.	1
SubSystemId	Reference ID for subsystem.	54677
SubSystemName	Name of the subsystem.	Coolant Pumps
ConstructionCommodityId	Reference ID for construction commodity.	32544
ConstructionCommodityName	Name of the construction commodity.	Galvanized Pipe
ComponentCharacteristics	Array of additional fields that are used to describe the component based on its component type. Each additional field will be provided by name and contain the value entered for the field for the component.	
Name	Field attribute name.	End Prep
Value	Field attribute values.	Beveled

## Sample JSON

```
{
  "ComponentsAndCharacteristics": [
    {
      "ComponentName": "2070-05-17-2019",
      "DisciplineId": 1,
      "DisciplineName": "Aggregates and Paving",
      "CommodityId": "",
      "CommodityName": "",
      "ComponentLevelId": 1,
      "ComponentLevelName": "Discipline",
      "ClaimingSchemeId": 16949,
      "ClaimingSchemeName": "Aggregates and Paving",
      "ProjectId": 159,
      "ProjectDisplay": "103114",
      "Description": "2070-05-17-2019",
      "Comments": "2070-05-17-2019",
      "ComponentQuantity": 10.00000,
      "ToDateQuantity": 0.00000,
      "UoMId": 82,
      "UoMName": "PLS",
      "WBS": "2070",
      "ScheduleId": "",
      "AGUGId": 147,
      "ConstructionAreaId": 22082,
      "ConstructionAreaName": "Roads/Parking",
      "Phase": "3",
      "SystemId": 22083,
      "SystemName": "20190101050551",
      "Line": "34",
    }
  ]
}
```

```
"TagNumber": "2244",
"Specification": "3434",
"CodeClass": "AA",
"MaterialCode": "001-AA",
"Size": "25",
"SizeCode": "004",
"Thickness": "2",
"Weight": "4",
"ITP": "",
"Supplier": "03904",
"ShopFieldId": 0,
"TurnoverId": 22084,
"TurnoverName": "20190101050551",
"TestPackageId": "",
"TurnoverComplete": "2019-05-17T05:00:00+00:00",
"TestPackageComplete": "2019-05-17T05:00:00+00:00",
"IsDefault": false,
"CostItemId": 218270,
"PercentageComplete": 0.00000,
"Id": 1250943,
"CreatedByDisplayId": "SUNIL007",
"CreatedDate": "2019-05-17T16:49:40.6532315+00:00",
"ModifiedByDisplayId": "SUNIL007",
"ModifiedDate": "2019-05-18T00:00:00+00:00",
"IsActive": true,
"AccountId": 1,
"SubSystemId": "",
"SubSystemName": "",
"ConstructionCommodityId": 660,
"ConstructionCommodityName": "Pipe curve",
"ComponentCharacteristics": [
  {
    "Name": "Actual Quantity",
    "Value": "10"
  },{
    "Name": "Asphalt Design",
    "Value": "PPRT"
  },{
    "Name": "Asphalt Mix",
    "Value": "Yes"
  },{
    "Name": "AUX 1",
    "Value": "Aux 1"
  },{
    "Name": "AUX 2",
    "Value": "Aux 2"
  },{
    "Name": "AUX 3",
    "Value": "Aux 3"
  },{
    "Name": "AUX 4",
    "Value": "Aux 4"
  },{
    "Name": "AUX 5",
    "Value": "Aux 5"
  },{
    "Name": "Designer",
```

```
"Value": "Yes"
}, {
  "Name": "Electrical Drawing",
  "Value": "No"
}, {
  "Name": "Engineer Responsible",
  "Value": "Yes"
}, {
  "Name": "Foundation Drawing",
  "Value": "Yes"
}, {
  "Name": "Mechanical Drawing",
  "Value": "Required"
}, {
  "Name": "Pipe Drawing",
  "Value": "No"
}
]
]
```



## Components

<b>Direction</b>		From the InEight cloud platform to external system.
<b>Frequency</b>		Data can be retrieved from InEight cloud platform at any time.
<b>Trigger Methods</b>		Determined by external system.
<b>Average Payload Size</b>		Tens of records. Maximum of 50 records per request.
<b>APIM Name</b>		Component_Get
<b>InEight Application</b>	<b>Starting Version</b>	23.6
	<b>Ending Version</b>	

<b>Direction</b>		From external system to the InEight cloud platform.
<b>Frequency</b>		Data can be retrieved from InEight cloud platform at any time.
<b>Trigger Methods</b>		Determined by external system.
<b>Average Payload Size</b>		Tens of records.
<b>APIM Name</b>		Component_Import
<b>InEight Application</b>	<b>Starting Version</b>	23.6
	<b>Ending Version</b>	

## Fields

**NOTE:**

Component must be given a discipline (activity) or commodity (material), but not both. The claiming scheme and the discipline/commodity value can be pulled from the WBS mapping in Plan, if given.

To update an existing component, the PlatformId must be provided. If the PlatformId is not provided, it will be considered as a new record.

Depth	Name	Type	Precision <sup>1</sup>	Parent	Req.
1	PlatformId	GUID	36		No
1	PlatformName	String	250		No
1	ProjectId	Number	Integer		No
1	ProjectDisplay	String	200		Yes
1	ComponentName	String	200		Yes
1	ComponentType	String	50		Yes
1	Description	String	250		No

Depth	Name	Type	Precision <sup>1</sup>	Parent	Req.
1	UoMName	String	255		No
1	DisciplineName	String	100		Yes*
1	CommodityName	String	100		Yes*
1	TagNumber	String	50		No
1	CreatedDate	String	Date		No
1	ModifiedDate	String	Date		No
1	IsActive	Boolean	NA		No
1	ContractId	Number	Integer		No
1	LineNumber	String	50		No
1	VendorName	String	250		No
1	ProcurementStatus	String	100		No
1	ScheduleActivityDisplayId	String	50		No
1	ScheduleActivityName	String	50		No
1	ScheduleStart	String	Date		No
1	ScheduleFinish	String	Date		No
1	WBSPHaseCode	String	250		No
1	ConstructionAreaName	String	250		No
1	SystemName	String	250		No
1	SubSystemPlatformId	String	36		No
1	SubSystemName	String	250		No
1	TurnoverName	String	250		No
1	ClaimingSchemeName	String	250		Yes
1	ComponentQuantity	Number	28,15		No
1	ConstructionCommodityName	String	250		No
1	Comments	String	250		No
1	ToDateQuantity	Number	28,15		No
1	Unit	String	250		No
1	BatteryLimit	String	250		No
1	ConstructionAreaPlatformId	String	36		No
1	Phase	String	250		No
1	Building	String	250		No

Depth	Name	Type	Precision <sup>1</sup>	Parent	Req.
1	Elevation	String	250		No
1	SystemPlatformId	String	36		No
1	Line	String	250		No
1	Specification	String	250		No
1	CodeClass	String	250		No
1	MaterialCode	String	250		No
1	PaintCode	String	250		No
1	Size	String	250		No
1	SizeCode	String	250		No
1	Thickness	String	250		No
1	Weight	String	250		No
1	ITP	String	250		No
1	Supplier	String	250		No
1	ShopFieldId	Number	Integer		No
1	ShopFieldName	String	250		No
1	RequisitionNumber	String	250		No
1	PO	String	250		No
1	PromiseDate	String	Date		No
1	RequiredDate	String	Date		No
1	MRR	String	250		No
1	LoadNumber	String	250		No
1	Laydown	String	250		No
1	TurnOverPlatformId	String	36		No
1	TurnoverComplete	String	Date		No
1	TestPackageComplete	String	Date		No
1	OwnerCode	String	250		No
1	CreatedByDisplayId	String	250		No
1	ModifiedByDisplayId	String	250		No
1	MM	String	250		No
1	IsDefault	Boolean	NA		No
1	PercentageComplete	String	28,15		No

Depth	Name	Type	Precision <sup>1</sup>	Parent	Req.
1	ConstructionCommodityPlatformId	String	36		No
1	ParentComponentPlatformId	String	36		No
1	ConstructionSegmentPlatformId	String	36		No
1	ConstructionSegmentname	String	250		No
1	ExternalUrl	String	250		No
1	EstimatingMaterialName	String	250		No
1	IsAssemblyLinked	Boolean	NA		No
1	ParentComponentName	String	250		No
1	TestPackageName	String	250		No
1	ComponentCharacteristics	Array	NA		No
2	Name	String	100	ComponentCharacteristics	No
2	Value	String	1024	ComponentCharacteristics	No
1	WorkPlanNumber	Array	NA		No
2	Items	Number	Integer	WorkPlanNumber	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 - 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
PlatformId	A unique ID for the record.	
PlatformName	InEight module or product name.	Plan
ProjectId	Internal Project ID reference.	103467
ProjectDisplay	Unique name or ID of a project that shows in the UI	103467
ComponentName	Name given to the component that shows in the UI.	Pipe Check sheet
ComponentType	Type of the component, such as Activity, Material, etc.	
Description	Public description of the component.	Checklist used for any pipe operations
UoMName	Unit of measure associated to the component.	
DisciplineName	Name of the discipline. Required if CommodityName is not provided.	Building

Name	Description	Example
CommodityName	Name of the commodity. Required if DisciplineName is not provided.	Cable
TagNumber	A unique tag number associated to the component.	TM654314J14
CreatedDate	Date when the record was created.	
ModifiedDate	Date on which the component record was last modified.	2019-05-14 13:07:31.0929640 +00:00
IsActive	Sending a value of false in this field causes the component record to be soft deleted from the InEight cloud platform. If a value is not provided, the default value true is used.	true
ContractId	Contract ID of the component.	
LineNumber	Contract line number of the associated contract to the component.	
Vendor	Name of the vendor associated with the Contract ID of the component.	
ProcurementStatus	Procurement status of the component.	
ScheduleActivityDisplayId	DisplayId of the associated schedule.	
ScheduleActivityName	Activity name of the associated schedule.	
ScheduleStart	Planned start date of the associated schedule.	
ScheduleFinish	Planned end date of the associated schedule.	
WBSPhaseCode	WBS Phase Code of the cost item associated with the component.	10.01.123.1234
ConstructionAreaName	Name of the construction area.	West End
SystemName	Name of the system.	Heating & Cooling
SubSystemPlatformId	SubSystemPlatformId of the Associated SubSystemPlatform to the Component.	
SubSystemName	Name of the subsystem.	Coolant Pumps
TurnoverName	Name of the turnover package.	Midtown Station
ClaimingSchemeName	Name of the claiming scheme.	Concrete
ComponentQuantity	Quantity used for measuring progress against work in the component.	1
ConstructionCommodityName	Name of the construction commodity.	Galvanized Pipe
Comments	Free-form text comments for the component.	Last updated op May 04
ToDateQuantity	Total claimed units of quantity at the component level.	1
Unit	Unit of the component.	

Name	Description	Example
BatteryLimit	Battery limit of the component.	
ConstructionAreaPlatformId	ConstructionAreaPlatformId of the associated ConstructionAreaPlatform to the component.	
Phase	Phase of the component.	
Building	Building of the component.	
Elevation	Elevation of the component.	
SystemPlatformId	SystemPlatformId of the associated system to the component.	
Line	Line of the component.	
Specification	Specification of the component.	
CodeClass	Code class of the component.	
MaterialCode	Material class of the component.	
PaintCode	Paint code of the component.	
Size	Size of the component.	
SizeCode	Size code of the component.	
Thickness	Thickness of the component.	
Weight	Wight of the component.	
ITP	ITP of the component.	
Supplier	Supplier of the component.	
ShopFieldName	Name of shop field.	54
Requisition Number	Value of the requisition number.	16
PO	Purchase order number.	1.1.1.1
Promise Date	Date promised for material.	2019-05-21 05:00:00.0000000 +00:00
Required Date	Date required for material delivery.	2019-05-21 05:00:00.0000000 +00:00
MRR	Material Receiving Report.	1A
LoadNumber	Load number of material.	123456
Laydown	Laydown area for material.	BOP 1A
TurnOverPlatformId	Internal reference ID for a turnover package.	34
TurnoverComplete	Turn over complete date.	2019-05-21 05:00:00.0000000 +00:00
TestPackageComplete	Test package complete date	2019-05-21 05:00:00.0000000 +00:00

Name	Description	Example
OwnerCode	Owner code of the component.	
CreatedByDisplayId	DisplayId for the user that created the component.	
ModifiedByDisplayId	DisplayId for the user that last modified the component.	
MM	MM of the component.	
IsDefault	Boolean value identifier to check whether the component is default or not.	False
PercentageComplete	Percentage of claim completed for the component.	100
ConstructionCommodityPlatformId	Reference ID for construction commodity.	
ParentComponentPlatformId	Reference ID for the parent component.	
ConstructionSegmentPlatformId	ID for the selected construction segment.	
ConstructionSegmentname	Name for the selected construction segment.	Block 1
ExternalUrl	URL value given to external component reference.	
EstimatingMaterialName	Internal component ID.	646744
IsAssemblyLinked	Date on which the component was created.	2019-05-14 13:07:31.0929640 +00:00
ParentComponentName	Name of the given parent component.	
TestPackageName	Name of the associated test package to the component.	
ComponentCharacteristics	Array of additional fields that are used to describe the component based on its component type. Each additional field will be provided by name and contain the value entered for the field for the component.	
Name	Field attribute name.	End Prep
Value	Field attribute values.	Beveled
WorkPlanNumber	Array of work plan numbers for the component.	
Items	Work plan numbers to associate with component	123456

## Sample JSON

```
[{
  "PlatformId": "909ad2ec-e9ed-4ea7-9938-403eb5c53e2a",
  "PlatformName": "PLAN",
  "ProjectId": 13044,
  "ProjectDisplay": "11052023",
  "ComponentName": "TestPack-9thJun001",
  "ComponentType": "Activity",
  "Description": "O9.81",
  "UoMName": "Barrel",
```

```

"DisciplineName": "Test Package",
"CommodityName": "",
"TagNumber": "09.81",
"CreatedById": 158219,
"CreatedDate": "2023-06-09T12:20:10.5171696+00:00",
"ModifiedById": 158219,
"ModifiedDate": "2023-06-09T12:20:10.5171696+00:00",
"IsActive": true,
"ContractId": 7200011144,
"LineNumber": "0002",
"VendorName": "Grow-Perini, A Joint Venture",
"ProcurementStatus": "09.81",
"ScheduleActivityDisplayId": "09.81",
"ScheduleActivityName": "09.81name",
"ScheduleStart": "06/08/2023 00:00:00",
"ScheduleFinish": "07/07/2023 00:00:00",
"WBSPhaseCode": "",
"ConstructionAreaName": "13044-12thMay-003",
"SystemName": "13044-12thMaySys-002",
"SubSystemId": 1042308,
"SubSystemPlatformId": "1f967d12-dfc7-45c0-a53e-e57be8c6fcac",
"SubSystemName": "13044-12thMaysubsystem-002",
"TurnoverName": "t3",
"ClaimingSchemeName": "Test Package",
"ComponentQuantity": 20.0,
"ConstructionCommodityName": "13044-12thMayConsComm-001",
"Comments": "09.81",
"ToDateQuantity": 0.0,
"Unit": "09.81",
"BatteryLimit": "09.81",
"ConstructionAreaPlatformId": "a642afe9-c4c6-43bc-8983-c486f0a384c7",
"Phase": "09.81",
"Building": "09.81",
"Elevation": "09.81",
"SystemPlatformId": "fbfeb5fe-b9d1-4e29-9539-f00781707832",
"Line": "09.81",
"Specification": "09.81",
"CodeClass": "09.81",
"MaterialCode": "09.81",
"PaintCode": "09.81",
"Size": "09.81",
"SizeCode": "09.81",
"Thickness": "09.81",
"Weight": "09.81",
"ITP": "09.81",
"Supplier": "09.81",
"ShopFieldId": 150,
"ShopFieldName": "Shop",
"RequisitionNumber": "09.81",
"PO": "09.81",
"PromiseDate": "2023-06-13T00:00:00+00:00",
"RequiredDate": "2023-06-22T00:00:00+00:00",
"MRR": "09.81",
"LoadNumber": "09.81",
"Laydown": "09.81",
"TurnOverPlatformId": "1b9d7558-ac11-42d6-bfdb-3114c3b3c11e",
"TurnoverComplete": "2023-06-22T00:00:00+00:00",

```



```
"TestPackageComplete": "",
"OwnerCode": "09.81",
"CreatedByDisplayId": "Santwana",
"ModifiedByDisplayId": "Santwana",
"MM": "09.81",
"IsDefault": false,
"PercentageComplete": 0.0,
"ConstructionCommodityPlatformId": "caaede76-f3ff-47f8-995c-c4d36991004a",
"ParentComponentPlatformId": "41a1fbe2-49de-4771-bd6e-1bd28b8af073",
"ConstructionSegmentId": 1042299,
"ConstructionSegmentPlatformId": "74e5d889-e5e0-497f-8e22-d470c43ecfe4",
"ConstructionSegmentName": "13044-12thMaySeg-001",
"ExternalUrl": "www.gmail.com",
"EstimatingMaterialName": "",
"IsAssemblyLinked": "",
"ParentComponentName": "Asm-1stJun-001",
"TestPackageName": "",
"ComponentCharacteristics": [{
  "Name": "AUX 1",
  "Value": "09.813"
}, {
  "Name": "AUX 2",
  "Value": "09.814"
}, {
  "Name": "AUX 3",
  "Value": "09.815"
}, {
  "Name": "Preliminary Status",
  "Value": "09.81"
}, {
  "Name": "KIX Asset Type",
  "Value": "09.827"
}, {
  "Name": "PLI Activity Type",
  "Value": "09.81"
}, {
  "Name": "Punch List ID",
  "Value": "09.81"
}],
"WorkPlanNumber": [80342]
}]
```

## Appendix A: Full Daily Plan Schema

Depth	Name	Type	Precision <sup>1</sup>	Parent
1	DailyPlan	Array		
2	<b>DailyPlanCostItem</b>	<b>Array</b>		<b>DailyPlan</b>
3	<b>DailyPlanCostItemEmployee</b>	<b>Array</b>		<b>DailyPlanCostItem</b>
4	ResourceId	String	50	DailyPlanCostItemEmployee
4	SubmittedDoubletimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	SubmittedOvertimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	SubmittedStandardHours	Decimal	5,2	DailyPlanCostItemEmployee
4	ApprovedDoubletimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	ApprovedOvertimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	ApprovedStandardHours	Decimal	5,2	DailyPlanCostItemEmployee
4	PlannedDoubletimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	PlannedOvertimeHours	Decimal	5,2	DailyPlanCostItemEmployee
4	PlannedStandardHours	Decimal	5,2	DailyPlanCostItemEmployee
4	EmployeeId	String	50	DailyPlanCostItemEmployee
4	CraftCode	String	50	DailyPlanCostItemEmployee
4	ReasonCodes	Array		DailyPlanCostItemEmployee
5	ReasonCodeId	String	50	ReasonCodes
5	Hour Type	String	50	ReasonCodes
5	Hours	Decimal	5,2	ReasonCodes
5	BillingCode	String	10	ReasonCodes
5	Premiums	List		ReasonCodes
3	<b>DailyPlanCostItemEquipment</b>	<b>Array</b>		<b>DailyPlanCostItem</b>
4	ResourceId	String	50	DailyPlanCostItemEquipment
4	ApprovedStandardHours	Decimal	5,2	DailyPlanCostItemEquipment
4	PlannedStandardHours	Decimal	5,2	DailyPlanCostItemEquipment
4	SubmittedStandardHours	Decimal	5,2	DailyPlanCostItemEquipment
4	EquipmentId	String	50	DailyPlanCostItemEquipment
4	EquipmentType	String	250	DailyPlanCostItemEquipment
4	ReasonCodes	Array		DailyPlanCostItemEquipment

Depth	Name	Type	Precision <sup>1</sup>	Parent
5	OperatedEmployeeId	String	50	ReasonCodes
5	ReasonCodeId	String	50	ReasonCodes
5	Hours	Number	5,2	ReasonCodes
5	BillingCode	String	10	ReasonCodes
<b>3</b>	<b>DailyPlanCostItemComponent</b>	<b>Array</b>		<b>DailyPlanCostItem</b>
4	ComponentId	Number	50	DailyPlanCostItemComponent
4	ComponentType	String	50	DailyPlanCostItemComponent
4	InstalledQuantity	Decimal	28,15	DailyPlanCostItemComponent
4	ProjectEstimatingResourceId	Number	50	DailyPlanCostItemComponent
4	ProjectEstimatingResourceSourceSystemId	String	50	DailyPlanCostItemComponent
4	ResourceId	Number	50	DailyPlanCostItemComponent
3	CostItemId	String	50	DailyPlanCostItem
3	TaskPriority	String	50	DailyPlanCostItem
3	PlanQuantity	Decimal	28,15	DailyPlanCostItem
3	SubmittedQuantity	Decimal	28,15	DailyPlanCostItem
3	ApprovedQuantity	Decimal	28,15	DailyPlanCostItem
3	CostItemSourceSystemId	String	50	DailyPlanCostItem
3	WBSCode	String	50	DailyPlanCostItem
<b>2</b>	<b>DailyPlanMaintenance</b>	<b>Array</b>		<b>DailyPlan</b>
3	EmployeeId	String	50	DailyPlanMaintenance
3	WorkOrderId	String	100	DailyPlanMaintenance
3	WBSPhaseCode	String	50	DailyPlanMaintenance
3	CostItemSourceSystemId	String	50	DailyPlanMaintenance
3	TotalHours	Number	16,5	DailyPlanMaintenance
3	MaintenanceEquipmentID	String	50	DailyPlanMaintenance
3	Segment1	String	50	DailyPlanMaintenance
3	Segment2	String	50	DailyPlanMaintenance
3	Segment3	String	50	DailyPlanMaintenance
3	Segment4	String	50	DailyPlanMaintenance
3	ReasonCodes	Array		DailyPlanMaintenance
4	ReasonCodeId	String	50	ReasonCodes

Depth	Name	Type	Precision <sup>1</sup>	Parent
4	Hours	Decimal	5,2	ReasonCodes
4	Premiums	List		ReasonCodes
<b>2</b>	<b>DailyPlanNote</b>	<b>Array</b>		<b>DailyPlan</b>
3	DailyPlanNoteEmployee	Array		DailyPlanNote
4	EmployeeDisplay	String	50	DailyPlanNoteEmployee
3	DailyPlanNoteEquipment	Array		DailyPlanNote
4	EquipmentId	String	50	DailyPlanNoteEquipment
3	DailyPlanNoteTag	Array		DailyPlanNote
4	TagCode	String	100	DailyPlanNoteTag
3	DailyPlanNoteTask	Array		DailyPlanNote
4	CostItemId	Number	50	DailyPlanNoteTask
3	Noteld	String	50	DailyPlanNote
3	Description	String	4000	DailyPlanNote
<b>2</b>	<b>DailyPlanSignoff</b>	<b>Array</b>		<b>DailyPlan</b>
3	EmployeeId	String	50	DailyPlanSignoff
3	QuestionnaireKey	String	50	DailyPlanSignoff
3	SignoffEmployeeId	String	50	DailyPlanSignoff
3	SignoffDate <sup>3</sup>	String	25	DailyPlanSignoff
3	SignoffStandardHours	Decimal	5,2	DailyPlanSignoff
3	SignoffOvertimeHours	Decimal	5,2	DailyPlanSignoff
3	SignoffDoubletimeHours	Decimal	5,2	DailyPlanSignoff
3	DailyPlanSignoffResponse	Array		DailyPlanSignoff
4	EmployeeId	String	50	DailyPlanSignoffResponse
4	QuestionKey	String	50	DailyPlanSignoffResponse
4	DailyPlanSignoffResponseId	String	50	DailyPlanSignoffResponse
4	ResponseKey	String	50	DailyPlanSignoffResponse
4	ResponseText	String	50	DailyPlanSignoffResponse
<b>2</b>	<b>DailyPlanSignins</b>	<b>Array</b>		<b>DailyPlan</b>
3	EmployeeDisplay	String	50	DailyPlanSignins
3	SignedInByEmployeeDisplay	String	50	DailyPlanSignins
3	SignedInByUserDisplay	String	50	DailyPlanSignins

Depth	Name	Type	Precision <sup>1</sup>	Parent
3	SignedInDate	String	25	DailyPlanSignins
4	DailyPlanSignInResponses	Array		DailyPlanSignins
4	DailyPlanSignInResponseId	String	50	DailyPlanSignInResponses
4	QuestionKey	String	50	DailyPlanSignInResponses
4	QuestionDescription	String	50	DailyPlanSignInResponses
4	ResponseKey	String	50	DailyPlanSignInResponses
4	ResponseDescription	String	50	DailyPlanSignInResponses
<b>2</b>	<b>DailyPlanBreaks</b>	<b>Array</b>		<b>DailyPlan</b>
3	EmployeeDisplay	String	50	DailyPlanBreaks
3	BreakNumber	Number	50	DailyPlanBreaks
3	BreakStartTime	String	25	DailyPlanBreaks
3	BreakInMinutes	Number	50	DailyPlanBreaks
<b>2</b>	<b>Executers</b>	<b>Array</b>		<b>DailyPlan</b>
3	RoleName	String	100	Executers
3	DisplayId	String	50	Executers
<b>2</b>	<b>Approvers</b>	<b>Array</b>		<b>DailyPlan</b>
3	RoleName	String	100	Approvers
3	DisplayId	String	50	Approvers
<b>2</b>	<b>DailyPlanEmployeePayrollIndicator</b>	<b>Array</b>		<b>DailyPlan</b>
3	DailyPlanEmployeePayrollIndicatorId	Number	50	DailyPlanEmployeePayrollIndicator
3	EmployeeId	String	50	DailyPlanEmployeePayrollIndicator
3	DailyPlanNoteId	Number	50	DailyPlanEmployeePayrollIndicator
3	PayrollIndicatorId	Number	50	DailyPlanEmployeePayrollIndicator
3	CostItemId	Number	50	DailyPlanEmployeePayrollIndicator
3	PayrollIdentifier	String	50	DailyPlanEmployeePayrollIndicator
3	WBSPhaseCode	String	50	DailyPlanEmployeePayrollIndicator
<b>2</b>	<b>DailyPlanClientSignoff</b>	<b>Array</b>		<b>DailyPlan</b>
3	Name	String	100	ClientSignoff
3	Email	String	100	ClientSignoff
2	PlanId <sup>2</sup>	String	50	DailyPlan
2	CreatedDate <sup>2</sup>	String	25	DailyPlan

Depth	Name	Type	Precision <sup>1</sup>	Parent
2	ProjectId	String	50	DailyPlan
2	LanguageKey	String	2	DailyPlan
2	Location	String	100	DailyPlan
2	CreatedById	String	50	DailyPlan
2	DeviceKey	String	50	DailyPlan
2	Shift	String	100	DailyPlan
2	PlanDate <sup>2</sup>	String	25	DailyPlan
2	PlanTitle	String	100	DailyPlan
2	PlanStatusCode	String	100	DailyPlan
2	ModifiedById	String	50	DailyPlan
2	ModifiedDate <sup>2</sup>	String	25	DailyPlan
2	PlannedDate <sup>2</sup>	String	25	DailyPlan
2	PlannedById	String	50	DailyPlan
2	ExecutedDate <sup>2</sup>	String	25	DailyPlan
2	ExecutedById	String	50	DailyPlan
2	ApprovedDate <sup>2</sup>	String	25	DailyPlan
2	ApprovedById	String	50	DailyPlan
2	ErrorHandlingToken	String	50	DailyPlan
2	CommitmentCode	String	100	DailyPlan
2	ExternalSyncStatus	Number	Integer	DailyPlan
2	CommitmentId	String	Integer	DailyPlan
2	ClientSignoffReportRecipients	List	1000	DailyPlan

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 - 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.