

Oracle Warehouse Management Cloud

WFM REST API Guide

26C



Contents

Get Help	i
2 Overview	3
Overview	3
3 WFM Entity Operations (POST)	5
WFM Entity Operations (POST)	5
4 WFM Entity Operations (GET)	25
WFM Entity Operations (GET)	25

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Some application pages have help icons  to give you access to contextual help. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons. If the page has contextual help, help icons will appear.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at [Oracle University](#).

Join Our Community

Use [Cloud Customer Connect](#) to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest [ideas](#) for product enhancements, and watch events.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to oracle_fusion_applications_help_ww_grp@oracle.com.

Thanks for helping us improve our user assistance!

2 Overview

Overview

This document is dedicated to the REST APIs that are supported by Oracle Workforce Management (WFM) Cloud. For more information about Restful Web Services, data input methodology, HTTP response, entity module, creating a resource using POST and obtaining resource representations using GET, refer WMS REST API Guide (navigate to *Logistics Cloud Suite* > Oracle Warehouse Management > WMS REST API Guide).

End User License Agreement

This guide contains REST APIs that are specific to Oracle Workforce Management (WFM) Cloud. These APIs should only be used by customers that have purchased WFM license.

3 WFM Entity Operations (POST)

WFM Entity Operations (POST)

This section details the POST entity operations for WFM.

SKU Lines

The **SKU Line** API allows you to create one or more SKU Lines in WFM.

URL

POST `..lgsfapi/v10/entity/sku_line/`

Fields

Name	Required	Type	Default	Description
name	X	String		SKU line name
description	X	String		SKU line description

Request Body Example (JSON)

Instance Create

```
{
  "fields": {
    {
      "name": "Skechers Shoes1234_23",
      "description": "Skechers Shoes for Women12345_23"
    }
  }
}
```

Bulk Create

```
{
  "options": {
    "commit_frequency": 2
  },
  "fields": [
    {
      "name": "SKU LINE PUMA",
      "description": "SKU PUMA T-SHIRT"
    },
    {
      "name": "SKU LINE PUMA10",
      "description": "SKU PUMA T-SHIRT10"
    }
  ]
}
```

```
}
```

Request Body Example (XML)

Instance Create

```
<request>
<fields>
<name>SKU PUMA SHOES</name>
<description>SKU PUMA SHOES FOR MEN</description>
</fields>
</request>
```

Bulk Create

```
<request>
<fields>
<list-item>
<name>SKU LINE School Uniform</name>
<description>SKU School Uniform Set for Kindergarden</description>
</list-item>
<list-item>
<name>SKU LINE School Uniform</name>
<description>SKU School Uniform Sets for 1 to 5th</description>
</list-item>
</fields>
<options>
<commit_frequency>2</commit_frequency>
</options>
</request>
```

Note: If the options parameter "commit_frequency" is not provided in the payload, the behavior defaults to same behavior as commit_frequency = 0, which is a all or nothing functionality. In other words, either all records are processed (if successful) or none of the records are processed even if 1 record fails validations.

Goals for System (WMS) Activities

The **Prod Goal Line Group** API allows you to create one or more goals for system activities in WFM.

URL

POST ..lgfapi/v10/entity/prod_goal_line_group/

Fields

Name	Required	Type	Description
facility_id	X	Integer	Facility ID
work_area_activity_id	X	Integer	WMS Internal ID of the Work Area Activity

sku_line_group_id	X	Integer	WMS Internal ID of the SKU Line Group
cal_month_id	X	Integer	Values 1 through 12. 1 for JAN, 2 for FEB..... 12 for DEC
cal_year	X	Integer	Four-digit year YYYY; Example 2023
uom_id	X	String	WMS Internal ID of the UOM
uom_per_hr	X	Integer	Qty expected to be handled in one hour (This is the goal)
is_pre_pack	X	Boolean	True or False; Indicates if the item represents a pre-pack

Request Body Example (JSON)

Instance Create

```
{
  "fields": {
    "facility_id": 648,
    "work_area_activity_id": 573,
    "sku_line_group_id": 22,
    "cal_month_id": 3,
    "cal_year": 2023,
    "uom_id": 2,
    "uom_per_hr": 200,
    "is_pre_pack": true
  }
}
```

Bulk Create

```
{
  "options": {
    "commit_frequency": 2
  },
  "fields": [
    {
      "facility_id": 648,
      "work_area_activity_id": 573,
      "sku_line_group_id": 22,
      "cal_month_id": 1,
      "cal_year": 2022,
      "uom_id": 1,
      "uom_per_hr": 750,
      "is_pre_pack": "FALSE"
    },
    {
      "facility_id": 648,
      "work_area_activity_id": 573,
      "sku_line_group_id": 22,
      "cal_month_id": 11,
      "cal_year": 2023,
      "uom_id": 1,
      "uom_per_hr": 750,
      "is_pre_pack": false
    }
  ]
}
```

```
},
{
  "facility_id": 648,
  "work_area_activity_id": 573,
  "sku_line_group_id": 22,
  "cal_month_id": 12,
  "uom_id": 1,
  "uom_per_hr": 750,
  "is_pre_pack": false
}
]
```

Request Body Example (XML)

Instance Create

```
<request>
  <fields>
    <facility_id>648</facility_id>
    <work_area_activity_id>553</work_area_activity_id>
    <sku_line_group_id>22</sku_line_group_id>
    <cal_month_id>1</cal_month_id>
    <cal_year>2027</cal_year>
    <uom_id>1</uom_id>
    <uom_per_hr>25</uom_per_hr>
    <is_pre_pack>FALSE</is_pre_pack>
  </fields>
</request>
```

Bulk Create

```
<request>
  <fields>
    <list-item>
      <facility_id>648</facility_id>
      <work_area_activity_id>475</work_area_activity_id>
      <sku_line_group_id>22</sku_line_group_id>
      <cal_month_id>1</cal_month_id>
      <cal_year>2023</cal_year>
      <uom_id>1</uom_id>
      <uom_per_hr>20</uom_per_hr>
      <is_pre_pack>FALSE</is_pre_pack>
    </list-item>
    <list-item>
      <facility_id>648</facility_id>
      <work_area_activity_id>292</work_area_activity_id>
      <sku_line_group_id>42</sku_line_group_id>
      <cal_month_id>2</cal_month_id>
      <cal_year>2026</cal_year>
      <uom_id>2</uom_id>
      <uom_per_hr>200</uom_per_hr>
      <is_pre_pack>TRUE</is_pre_pack>
    </list-item>
  </fields>
</request>
```

Note: If the options parameter "commit_frequency" is not provided in the payload, the behavior defaults to same behavior as commit_frequency = 0, which is a all or nothing functionality. In other words, either all records are processed (if successful) or none of the records are processed even if 1 record fails validations.

Work Area Activities

The **Work Area Activity** API allows you to create one or more WMS, Non-WMS, or Ad hoc activities in WFM

URL

POST `..lgfapi/v10/entity/work_area_activity/`

Fields

Name	Required	Type	Description
facility_id	X	Integer	Facility ID
code	X	String	Code of the WMS, Non-WMS or Ad hoc Work Area Activity.
name	X	String	Name of the WMS, Non-WMS or Ad hoc Work Area Activity.
work_area_activity_type_id	X	Integer	WFM Internal ID of the type of Work Area Activity. The following IDs are supported: 0 – Represents WMS Activity 1 – Represents Non-WMS Activity 2 – Represents Ad hoc Activity
work_area_id	X	integer	WFM Internal ID of Work Area associated to the WMS, Non-WMS or Ad hoc Activity.

Request Body Example (JSON)

Instance Create

```
{
  "fields": {
    "facility_id": 648,
    "code": "IBACT01",
    "name": "Inbound Receiving Activity",
    "work_area_activity_type_id": 0,
    "work_area_id": 18
  }
}
```

Bulk Create

```
{
```

```
"options":
{
  "commit_frequency": 2
},

"fields": [
{
  "facility_id": 648,
  "code": "IBACT01",
  "name": "Inbound Receiving Activity",
  "work_area_activity_type_id": 0,
  "work_area_id": 18
},

{
  "facility_id": 648,
  "code": "OBVAS01",
  "name": "Outbound VAS Activity",
  "work_area_activity_type_id": 1,
  "work_area_id": 21
}
]
}
```

Request Body Example (XML)

Instance Create

```
<request>
  <fields>
    <facility_id>648</facility_id>
    <code>IBACT01</code>
    <name>Inbound Receiving Activity</name>
    <work_area_activity_type_id>0</work_area_activity_type_id>
    <work_area_id>18</work_area_id>
  </fields>
</request>
```

Bulk Create

```
<request>
  <fields>
    <list-item>
      <facility_id>648</facility_id>
      <code>IBACT01</code>
      <name>Inbound Receiving Activity</name>
      <work_area_activity_type_id>0</work_area_activity_type_id>
      <work_area_id>18</work_area_id>
    </list-item>
    <list-item>
      <facility_id>648</facility_id>
      <code>MISCACT</code>
      <name>Maintenance Activities</name>
      <work_area_activity_type_id>2</work_area_activity_type_id>
      <work_area_id>27</work_area_id>
    </list-item>
  </fields>
  <options>
    <commit_frequency>2</commit_frequency>
  </options>
</request>
```

Note: If the options parameter "commit_frequency" is not provided in the payload, the behavior defaults to same behavior as commit_frequency = 0, which is an all or nothing functionality. In other words, either all records are processed (if successful) or none of the records are processed even if 1 record fails validations.

Map WMS Screens to WMS Activities

The **WAA Screen XREF** API allows you to associate one or more WMS Screens to WMS activities in WFM.

URL

POST ../lgfapi/v10/entity/waa_screen_xref/

Fields

Name	Required	Type	Description
work_area_activity_id	X	Integer	WFM Internal ID of the WMS Activity
module_instance_id	X	Integer	WMS Internal ID of the WMS Screen

Request Body Example (JSON)

Instance Create

```
{
  "fields": {
    "work_area_activity_id": 12,
    "module_instance_id": 18
  }
}
```

Bulk Create

```
{
  "options": {
    "commit_frequency": 2
  },
  "fields": [
    {
      "work_area_activity_id": 12,
      "module_instance_id": 18
    },
    {
      "work_area_activity_id": 12,
      "module_instance_id": 25
    }
  ]
}
```

Request Body Example (XML)

Instance Create

```
<request>
  <fields>
    <work_area_activity_id>12</work_area_activity_id>
    <module_instance_id>18</module_instance_id>
  </fields>
</request>
```

Bulk Create

```
<request>
  <fields>
    <list-item>
      <work_area_activity_id>12</work_area_activity_id>
      <module_instance_id>18</module_instance_id>
    </list-item>
    <list-item>
      <work_area_activity_id>12</work_area_activity_id>
      <module_instance_id>25</module_instance_id>
    </list-item>
  </fields>
  <options>
    <commit_frequency>2</commit_frequency>
  </options>
</request>
```

Note: If the options parameter "commit_frequency" is not provided in the payload, the behavior defaults to same behavior as commit_frequency = 0, which is a all or nothing functionality. In other words, either all records are processed (if successful) or none of the records are processed even if 1 record fails validations.

Clock

The **Clock** API allows you to create one or more clock records in WFM. The data is posted to the stage view (Clock Stage tab of Input Interfaces) and the process to move data to the main view (Clock Record) is either synchronous or asynchronous depending on the `async_flg` on the API Header.

URL

```
POST ../lgfapi/v10/stage/clock_record/
```

Request Body

- **async_flg** - true/false; default true
 - When false:
 - Posts data to Stage Record of Clock Input Interface and the process to move data to Clock Record is triggered immediately after.
 - If all records are processed successfully, response 200 OK is returned with a response body which indicates that all records have been processed successfully. All the records show up in Clock Record view.
 - If one or more records fail validations, response 400 “Bad Request” is returned with the count of failed records in the response. Since clock records need to be sequential and are dependent on other clock records (such as facility clock-in before detail clock-in), no records will be processed after the first failure has been encountered for an employee.

Note: It is advisable not to use `async_flg = false` if you intend to post large volume of records since the calling system may have to wait for considerable time to receive a response from WFM.

- When true:
 - Posts data to Stage Record of Clock Input Interface
 - Returns response 200 OK with a response body which indicates that records have been submitted for further processing.
 - Records in stage table are processed asynchronously and if there are no errors, all records eventually show up in Clock Record view.
- **header**
 - document_version
 - origin_system
 - client_env_code
 - parent_company_code
 - entity - "clock_record"
 - timestamp
 - facility_code
 - company_code
 - messageid
- **stage_clock_record_list** - The table below describes the fields within this parameter:

Name	Required	Type	Description
action_code	X	String	CREATE/UPDATE
cw_user	X	String	Username of the employee that needs to be clocked
ci_ts	C	DateTime	Clock in timestamp

			Format: YYYYMMDDHHMMSS Example: 20260325081020
co_ts	C	DateTime	Clock out timestamp Format: YYYYMMDDHHMMSS Example: 20260325183050
cico_entity	X	String	Valid Values: FACILITY, BREAK, NONWMS, ADHOC
cw_break	C	String	Name of the break taken by employee. Required when cico_entity is BREAK
work_area_activity	C	String	Work Area Activity Code of the Non-WMS or Ad hoc activity. Required when cico_entity is NONWMS or ADHOC
handling_entity	C	String	See note below under "Handling Attributes for Non-WMS Activities"
handling_qty	C	Integer	See note below under "Handling Attributes for Non-WMS Activities"
handling_uom	C	String	See note below under "Handling Attributes for Non-WMS Activities"
wfm_item_category	O	String	See note below under "Handling Attributes for Non-WMS Activities"
lpn_type	O	String	See note below under "Handling Attributes for Non-WMS Activities"

where X = Mandatory; O = Optional; C = Conditional

Handling Attributes for Non-WMS Activity

The fields `handling_entity`, `handling_qty`, `handling_uom`, `wfm_item_category` and `lpn_type` are used when `cico_entity` is NONWMS and will be accepted only under the following conditions:

- A CREATE that has both `ci_ts` and `co_ts`.
- An UPDATE that has only `co_ts`.
- If an UPDATE is sent that has both `ci_ts` and `co_ts`, it is considered a case of exact match where WFM will attempt to find an existing clock record based on `work_area_activity`, `ci_ts` and `co_ts` and update the `handling_entity`, `handling_qty`, `handling_uom`, `wfm_item_category` and `lpn_type` of the record.
- If an UPDATE is sent that has both `ci_ts` and `co_ts`, fields `handling_entity`, `handling_qty` and `handling_uom` are mandatory.
- `handling_entity` must be "ITEM", "LPN" or "PALLET".
- `handling_uom` should be an existing UOM in WMS.

- wfm_item_category is optional and can be sent only if handling_entity is "ITEM". If sent it must exist in WFM Item Category.
- lpn_type is optional and can be sent only if handling_entity is "LPN" or "PALLET". If sent, it must exist in LPN Type.

Request Body Example (JSON)

Example with all Fields

```
{
  "async_flg": false,
  "header": {
    "document_version": "26B",
    "origin_system": "QA",
    "client_env_code": "ABC",
    "parent_company_code": "ABC",
    "entity": "clock_record",
    "timestamp": "2026-01-23T12:12:12",
    "facility_code": "QATST01",
    "company_code": "QATSTPC",
    "messageid": 12345
  },
  "stage_clock_record_list": [{
    "action_code": "CREATE",
    "cw_user": "KKALL01",
    "cico_entity": "FACILITY",
    "ci_ts": "2026-01-23T19:01:29",
    "co_ts": "",
    "cw_break": "",
    "work_area_activity": "",
    "handling_entity",
    "wfm_item_category": "",
    "lpn_type": "",
    "handling_uom": "",
    "handling_qty": ""
  }]
}
```

Example with all the fields

```
{
  "async_flg": false,
  "header": {
    "document_version": "26B",
    "origin_system": "QA",
    "client_env_code": "ABC",
    "parent_company_code": "ABC",
    "entity": "clock_record",
    "timestamp": "2026-01-23T12:12:12",
    "facility_code": "QATST01",
    "company_code": "QATSTPC",
    "messageid": 12345
  },
  "stage_clock_record_list": [{
    "action_code": "CREATE",
    "cw_user": "KKALL01",
    "cico_entity": "FACILITY",
    "ci_ts": "2026-01-23T19:01:29"
  }]
}
```

Employee Location Track Data

There are two APIs that can be used to post employee location track data to WFM. A single object upload REST API to post single records directly to the main view in WFM and a bulk upload REST API to post records in bulk that are first written to stage view and then moved to main view either synchronously or asynchronously.

Single Object Upload REST API

This API can be used to post single employee location track record to WFM. The data is posted synchronously to the main view (Employee Location Track) in WFM. Multiple records cannot be posted through this API.

URL

POST `..lgfapi/v10/entity/employee_location_track/`

Fields

Name	Required	Type	Description
company_id	X	Integer	Company ID
facility_id	X	Integer	Facility ID
cw_user_id	X	Integer	WMS Internal ID of the employee
location_barcode	C	String	Barcode of the location defined in WMS locations UI. Either this should be sent OR XYZ Coordinates have to be sent
x_coordinate	C	Integer	These should be sent if barcode is not sent. Field supports upto 38 digits regardless of decimal position. Maximum decimals is governed by the configuration in the facility for location coordinates, rounded up to decimals allowed. For example, if the decimal setting on location coordinates is 4 and the value sent in the payload is 123.456789, the value will be rounded to 123.4568
y_coordinate	C	Integer	
z_coordinate	C	Integer	
Coordinate UOM	O	String	UOM of the XYZ coordinates. This field is not mandatory but if sent, it should be a valid UOM code defined in WMS as a DIMENSION UOM class. Also, if sent, the Location Coordinates Base UOM should be configured on the facility to perform conversion.
tracked_timestamp	X	DateTime	Time at which the barcode was scanned or XYZ coordinate of the user was captured. Can be sent with or without time zone component. Example without time zone: "2024-08-07T00:00:00" Example with time zone: "2024-08-07T00:00:00+05:30"

			• If sent without time zone component, WMS assumes the tracked_timestamp to be in the time zone of the facility in the payload and the tracked_timestamp is used as-is. • If sent with time zone component, the time zone is used to convert tracked_timestamp to the facility time zone. Note: If sending a time zone component, ensure that the requesting user's default facility's time zone is same as the time zone of the facility in the payload, else the conversion may result in undesirable results Example 1: ◦ User's default facility time zone: +05:30 ◦ Time zone of the facility_id passed in the payload: +05:30 ◦ Tracked_timestamp sent with any time zone is converted appropriately. Example 2: ◦ User's default facility time zone: +08:00 ◦ Time zone of the facility_id passed in the payload: +05:30 ◦ Tracked_timestamp sent with any time zone is NOT converted appropriately
screen_name	X	String	Any text that identifies the app, device or screen in the external system that was used to capture user's location
equipment_type	O	String	If sent, it can be any text that identifies the equipment the user was operating when the location was captured by the external system.
task_nbr	O	String	If sent, the task nbr has to match task nbr in WMS Task UI
task_type_exec_seq_nbr	O	Integer	Represents the execution seq nbr of WMS task If the external system has this information, it can be sent.

where X = Mandatory; O = Optional; C = Conditional

Request Body Example (JSON)

With location_barcode

```
{
  "fields": {
    "company_id": 648,
    "facility_id": 722
    "cw_user_id" : 3452,
    "location_barcode": "R010503",
    "tracked_timestamp": "2024-08-07T00:00:00",
    "screen_name": "RFID Receiver 1"
  }
}
```

With xyz coordinates

```
{
  "fields": {
```

```
"company_id": 648,  
"facility_id": 722  
"cw_user_id" : 3452,  
"x_coordinate": 3,  
"y_coordinate": 4,  
"z_coordinate": 1,  
"coordinate_uom_code" 'M'  
"tracked_timestamp": "2024-08-07T00:00:00",  
"screen_name": "RFID Receiver 1"  
}  
}
```

With all fields

```
{  
  "fields": {  
    "company_id": 648,  
    "facility_id": 722  
    "cw_user_id" : 3452,  
    "x_coordinate": 3,  
    "y_coordinate": 4,  
    "z_coordinate": 1,  
    "coordinate_uom_code" 'M'  
    "tracked_timestamp": "2024-08-07T00:00:00",  
    "screen_name": "RFID Receiver 1",  
    "equipment_type": "PLTJK",  
    "task_nbr": "TASK00300102",  
    "task_type_exec_seq_nbr": 1  
  }  
}
```

Request Body Example (XML)

```
<request>  
  <fields>  
    <company_id>648</company_id>  
    <facility_id>722</facility_id>  
    <cw_user_id>3452</cw_user_id>  
    <location_barcode>R010503</location_barcode>  
    <tracked_timestamp>2024-08-07T00:00:00</tracked_timestamp>  
    <screen_name>RFID Receiver 1</screen_name>  
  </fields>  
</request>
```

Bulk Upload REST API

This API can be used to post single or multiple employee location track records to WFM. The data is posted to the stage view (Employee Location Track Stage tab of Input Interfaces) and the process to move data to the main view (Employee Location Track) is either synchronous or asynchronous depending on the `async_flg` on the API Header.

Note: This API supports only JSON format. XML is not supported.

URL

POST `..lgfapi/v10/stage/employee_location_track/`

Request Body

- **async_flg** - true/false; default true
 - When false:
 - Posts data to the Stage tab of Employee Location Track Input Interface and the process to move data to Employee Location Track is triggered immediately after.
 - If all records are processed successfully, a 204 is returned with a response body which indicates that all records have been processed successfully. All the records show up in Employee Location Track view.
 - If one or more records fail validations, a 204 is returned with a response body which indicates the number of records that had errors. Failed records remain in the Stage tab with status marked as "Failed". Records that go through successful validations show up in the Employee Location Track view.
- **Note:** It is advisable not to use `async_flg = false` if you intend to post large volume of records since the calling system may have to wait for considerable time to receive a response from WFM.
- When true:
 - Posts data to the Stage tab of Employee Location Track Input Interface
 - Returns 201 with a response body which indicates that the request has been received successfully and submitted for further processing.
 - The process to move data to Employee Location Track is triggered ONLY after sending the response to the caller.
- **header**
 - `origin_system`
 - `client_env_code`
 - `parent_company_code`
 - `entity` - "stage_employee_location_track"
 - `timestamp`
 - `facility_code`
 - `company_code`
 - `messageid`
- **stage_employee_location_track_list** - The table below lists the attributes within this parameter:

Name	Requi	Type	Description
cw_ user	X	String	Employee whose location is tracked
locatio barcod	C	String	Barcode of the location defined in WMS locations UI. Either this should be sent OR XYZ Coordinates have to be sent
x_ coordi	C	Intege	These should be sent if barcode is not sent. Field supports upto 38 digits regardless of decimal position. Maximum decimals is governed by the configuration in the facility for location coordinates, rounded up to decimals allowed. For example, if the decimal setting on location coordinates is 4 and the value sent in the payload is 123.456789, the value will be rounded to 123.4568

y_ coord	C	Intege	
z_ coord	C	Intege	
coord uom_ code	O	String	UOM of the XYZ coordinates. This field is not mandatory but if sent, it should be a valid UOM code defined in WMS as a DIMENSION UOM class. Also, if sent, the Location Coordinates Base UOM should be configured on the facility to perform conversion.
tracked_ timestamp	X	DateT	Time at which the barcode was scanned or XYZ coordinate of the user was captured. Can be sent with or without time zone component. Example without time zone: "2024-08-07T00:00:00" Example with time zone: "2024-08-07T00:00:00+05:30" ○ If sent without time zone component, WMS assumes the tracked_timestamp to be in the time zone of the facility in the payload and the tracked_timestamp is used as-is. ○ If sent with time zone component, the time zone is used to convert tracked_timestamp to the facility time zone. Note: If sending a time zone component, ensure that the requesting user's default facility's time zone is same as the time zone of the facility facility in the payload, else the conversion may result in undesirable results. Example 1: - User's default facility time zone: +05:30 - Time zone of the facility_code passed in the payload: +05:30 - Tracked_timestamp sent with any time zone is converted appropriately. Example 2: - User's default facility time zone: +08:00 - Time zone of the facility_code passed in the payload: +05:30 - Tracked_timestamp sent with any time zone is NOT converted appropriately
screen_ name	X	String	Any text that identifies the app, device or screen in the external system that was used to capture user's location
equipment_ type	O	String	If sent, it can be any text that identifies the equipment the user was operating when the location was captured by the external system.
task_ nbr	O	String	If sent, the task nbr has to match task nbr in WMS Task UI
task_ type_ exec_	O	Intege	Represents the execution seq nbr of WMS task If the external system has this information, it can be sent.

seq_			
nbr			

where X = Mandatory; O = Optional; C = Conditional

Request Body Example (JSON) - With all fields

```
{
  "async_flg":false,
  "header": {
    "document_version": "24D",
    "origin_system": "QA",
    "client_env_code": "",
    "parent_company_code": "",
    "entity": "stage_employee_location_track",
    "timestamp": "2024-07-23T12:12:12",
    "facility_code": "QATST01",
    "company_code": "QATSTPC",
    "messageid": 12345
  },
  "stage_employee_location_track_list": [{
    "cw_user": "KKALL01",
    "location_barcode": "R010503",
    "x_coordinate": 0,
    "y_coordinate": 0,
    "z_coordinate": 0,
    "coordinate_uom_code": "",
    "tracked_timestamp": "2024-06-12T19:01:29",
    "screen_name": "emp track app",
    "equipment_type": "",
    "task_nbr": "",
    "task_type_exec_seq_nbr": 0
  }]
}
```

Request Body Example (JSON) - With required fields

```
{
  "async_flg":false,
  "header": {
    "document_version": "24D",
    "origin_system": "QA",
    "client_env_code": "",
    "parent_company_code": "",
    "entity": "stage_employee_location_track",
    "timestamp": "2024-07-23T12:12:12",
    "facility_code": "QATST01",
    "company_code": "QATSTPC",
    "messageid": 12345
  },
  "stage_employee_location_track_list": [{
    "cw_user": "KKALL01",
    "location_barcode": "R010503",
    "tracked_timestamp": "2024-06-12T19:01:29",
    "screen_name": "emp track app"
  }]
}
```

WFM Item Category

The WFM Item Category API allows you to create one or more item categories in WFM.

URL

```
POST ..lgfapi/v10/entity/wfm_item_category/
```

Fields

WFM Item Category

Name	Required	Type	Description
company_id	X	Integer	Company ID
code	X	String	Code of the WFM Item Category
description	O	String	Description of WFM Item Category

where X = Mandatory; O = Optional; C = Conditional

Request Body Example (JSON)

Instance Create

```
{
  "fields": {
    "company_id": 369,
    "code" : "BNWFMCAT102",
    "description": "Machinery 102"
  }
}
```

Bulk Create

```
{
  "fields": [
    {
      "company_id": 369,
      "code" : "BNWFMITMCAT01",
      "description": "BN WFM Item Category 1"
    },
    {
      "company_id": 369,
      "code" : "BNWFMITMCAT02",
      "description": "BN WFM Item Category 2"
    },
    {
      "company_id": 369,
      "code" : "BNWFMITMCAT03",
      "description": "BN WFM Item Category 3"
    }
  ]
}
```

```
}
```

Bulk Create with commit frequency

```
{
  "options": {
    "commit_frequency": 2
  },
  "fields": [
    {
      "company_id": 369,
      "code" : "BNWFMITMCAT01",
      "description": "BN Item Category 1"
    },
    {
      "company_id": 369,
      "code" : "BNWFMITMCAT02",
      "description": "BN Item Category 2"
    }
  ]
}
```

Request Body Example (XML)

Instance Create

```
<request>
  <fields>
    <company_id>369</company_id>
    <code>BNITMCAT03</code>
    <description>BN Item Category 3</description>
  </fields>
</request>
```

Bulk Create

```
<request>
  <fields>
    <list-item>
      <company_id>369</company_id>
      <code>BNWFMITMCAT04</code>
      <description>BN Item Category 4</description>
    </list-item>
    <list-item>
      <company_id>369</company_id>
      <code>BNWFMITMCAT02</code>
      <description>BN Item Category 2</description>
    </list-item>
  </fields>
</request>
```

Bulk Create with commit frequency

```
<request>
  <fields>
    <list-item>
      <company_id>369</company_id>
      <code>BNWFMITMCAT01</code>
      <description>Item Category 1</description>
    </list-item>
    <list-item>
      <company_id>369</company_id>
      <code>BNWFMITMCAT06</code>
      <description>Item Category 06</description>
    </list-item>
```

```
</fields>
<options>
  <commit_frequency>2</commit_frequency>
</options>
</request>
```

4 WFM Entity Operations (GET)

WFM Entity Operations (GET)

This section overviews the GET Entity Operations for WFM.

WMS Activity Entities

The WMS Activity view captures activities performed by users on the warehouse floor using Oracle Cloud Warehouse Management System (WMS). The following APIs can be used to get relevant **WMS Activity** data.

Note: Operations performed using all WMS RF Transactions are written to WMS Activity. As of Release 24D, user activity information can also be posted through some of the operational WMS REST APIs. Refer WFM User Guide (navigate to *Logistics Cloud Suite* > Oracle Warehouse Management > Books > WFM User Guide) for the list of these APIs.

WMS Activity URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/wms_activity/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/wms_activity/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/wms_activity/{id}
```

WMS Activity Dtl URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/wms_activity_dtl/{id}
```

Fetch paginated results when users use the following URL:

```
GET .../lgfapi/v10/entity/wms_activity_dtl/
```

Fetch non-paginated result by specific 'ID'

```
GET .../lgfapi/v10/entity/wms_activity_dtl/{id}
```

WMS Activity Code URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/wms_activity_code/{id}
```

Fetch paginated results when users use the following URL:

```
GET .../lgfapi/v10/entity/wms_activity_code/
```

Fetch non-paginated result by specific 'ID'

```
GET .../lgfapi/v10/entity/wms_activity_code/{id}
```

WMS Activity Status URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/wms_activity_status/{id}
```

Fetch paginated results when users use the following URL:

```
GET .../lgfapi/v10/entity/wms_activity_status/
```

Fetch non-paginated result by specific 'ID'

```
GET .../lgfapi/v10/entity/wms_activity_status/{id}
```

WMS Activity Track Entities

As a first step, the **WMS WFM Interface** scheduled job moves WMS data from **WMS Activity** to **WMS Activity Track** view. The following APIs can be used to get relevant **WMS Activity Track** data.

WMS Activity Track URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/wms_activity_track/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/wms_activity_track/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/wms_activity_track/{id}
```

WMS Activity Track Dtl URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/wms_activity_track_dtl/{id}
```

Fetch paginated results when users use the following URL:

```
GET .../lgfapi/v10/entity/wms_activity_track_dtl/
```

Fetch non-paginated result by specific 'ID'

```
GET .../lgfapi/v10/entity/wms_activity_track_dtl/{id}
```

WMS Activity Track Status URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/wms_activity_track_status/{id}
```

Fetch paginated results when users use the following URL:

```
GET .../lgfapi/v10/entity/wms_activity_track_status/
```

Fetch non-paginated result by specific 'ID'

```
GET .../lgfapi/v10/entity/wms_activity_track_status/{id}
```

Daily Activity Track Entities

As a second step, the **WMS WFM Interface** scheduled job moves WMS data from **WMS Activity Track** to **Daily Activity Track** view. The following APIs can be used to get relevant **Daily Activity Track** data.

Daily Activity Track URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/daily_activity_track/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/daily_activity_track/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/daily_activity_track/{id}
```

Daily Activity Track Dtl URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/daily_activity_track_dtl/{id}
```

Fetch paginated results when users use the following URL:

```
GET .../lgfapi/v10/entity/daily_activity_track_dtl/
```

Fetch non-paginated result by specific 'ID'

```
GET .../lgfapi/v10/entity/daily_activity_track_dtl/{id}
```

Clock Record

The Clock Record view captures employee clock data. The following APIs can be used to get relevant Clock Record data.

Clock Record Hdr URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/clock_record_hdr/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/clock_record_hdr/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/clock_record_hdr/{id}
```

Clock Record Dtl URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/clock_record_dtl/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/clock_record_dtl/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/clock_record_dtl/{id}
```

Stage Clock Record

Employee clock data that is interfaced from external systems using Clock input interface or Clock REST API is first written to Stage Clock Record before validating and moving to Clock Record. The following APIs can be used to get relevant Stage Clock Record data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/stage_clock_record/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_clock_record/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_clock_record/{id}
```

Employee Location Track

The **Employee Location Track** view provides a consolidated list of all employee location track records that have been captured during WMS RF location scans or interfaced/posted through input interface or REST APIs. The following APIs can be used to get relevant **Employee Location Track** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/employee_location_track/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/employee_location_track/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/employee_location_track/{id}
```

Stage Employee Location Track

Employee Location Track data that is interfaced from external systems using the **Employee Location Track** input interface OR the bulk upload **stage_employee_location_track** REST API is first written to **Stage Employee Location Track** before validating and moving to **Employee Location Track**. The following APIs can be used to get relevant **Stage Employee Location Track** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/stage_employee_location_track/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_employee_location_track/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_employee_location_track/{id}
```

Employee Travel Time

The **Employee Travel Time** view provides the consolidated time taken by employees to perform a certain transaction that required traveling to multiple locations. The following APIs can be used to get relevant **Employee Travel Time** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/employee_travel_time/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/employee_travel_time/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/employee_travel_time/{id}
```

Daily Travel Time

The **Daily Travel Time** view provides the total time taken by employees to perform a certain transaction throughout a day. The following APIs can be used to get relevant **Daily Travel Time** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/daily_travel_time/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/daily_travel_time/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/daily_travel_time/{id}
```

Travel Time Entity Status

The **Travel Time Entity Status** is used to define status of both **Employee Location Track** and **Employee Travel Time** . The following APIs can be used to get relevant **Employee Location Track** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/travel_time_entity_status/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/travel_time_entity_status/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/travel_time_entity_status/{id}
```

Team

A **Team** consists of groups of employees. The following APIs can be used to get relevant **Team** information from WFM.

Note: As of Release 26B there is no functionality in WFM involving teams, however it could be used in future.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/team/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/team/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/team/{id}
```

Team User XREF

A **Team** consists of group of employees. The following APIs can be used to get the association between **Users** and **Teams** in WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/team_user_xref/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/team_user_xref/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/team_user_xref/{id}
```

SKU Line

SKU Lines area assigned to items in WMS in the items view. The same SKU Lines are also defined in WFM that are captured in the **SKU Line** view. The following APIs can be used to get relevant **SKU Line** data from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/sku_line/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/sku_line/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/sku_line/{id}
```

Stage SKU Line

SKU Line data that is interfaced from external systems using SKU Line input interface is first written to **Stage SKU Line** before validating and moving to **SKU Line**. The following APIs can be used to get relevant **Stage SKU Line** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/stage_sku_line/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_sku_line/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_sku_line/{id}
```

SKU Line Group

SKU Lines Groups are item categories against which goals for WMS activities can be defined. The following APIs can be used to get relevant **SKU Line Group** from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/sku_line_group/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/sku_line_group/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/sku_line_group/{id}
```

SKU Line Group XREF

Multiple SKU Lines are grouped into a single SKU Line Group. The following APIs can be used to get the association between **SKU lines** and **SKU Line Groups** in WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/sku_line_group_xref/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/sku_line_group_xref/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/sku_line_group_xref/{id}
```

Cost Center

There could be a single or multiple cost centers within a facility which is captured in the **Cost Center** view. The following APIs can be used to get relevant **Cost Center** data from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/cost_center/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/cost_center/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/cost_center/{id}
```

Work Group

There could be a single or multiple work groups within each cost center which is captured in the **Work Group** view. The following APIs can be used to get relevant **Work Group** data from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/work_group/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/work_group/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/work_group/{id}
```

Work Area

Users are assigned to different work areas which is captured in the **Work Area** view. The following APIs can be used to get relevant **Work Area** information from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/work_area/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/work_area/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/work_area/{id}
```

Work Area Activity Entities

All activities that are expected to be performed by users on the warehouse floor is defined in the **Work Area Activity** view. The activities are categorized as WMS, Non-WMS, Ad hoc activities. The following APIs can be used to get relevant **Work Area Activity** information from WFM.

Work Area Activity URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/work_area_activity/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/work_area_activity/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/work_area_activity/{id}
```

Work Area Activity Type URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/work_area_activity_type/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/work_area_activity_type/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/work_area_activity_type/{id}
```

Stage Work Area Activity

WMS, Non-WMS, Ad hoc Activities are interfaced from external systems using the **Work Area Activity** input interface is first written to **Stage Work Area Activity** before validating and moving to **Work Area Activity**. The following APIs can be used to get relevant **Work Area Activity** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/stage_work_area_activity/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_work_area_activity/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_work_area_activity/{id}
```

WAA Screen XREF

Any activity performed using WMS RF screen is mapped to a WMS activity in WFM. The mapping between WMS RF screens and WMS activities is captured in WAA Screen XREF. The following APIs can be used to get relevant **WAA Screen XREF** information from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/waa_screen_xref/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/ waa_screen_xref /
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/ waa_screen_xref/{id}
```

Stage WAA Screen XREF

The mapping of WMS Screens to WMS Activities that is interfaced from external systems using the **WAA Screen XREF** input interface is first written to **Stage WAA Screen XREF** before validating and moving to **WAA Screen XREF**. The following APIs can be used to get relevant **WAA Screen XREF** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/stage_waa_screen_xref/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_waa_screen_xref/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_waa_screen_xref/{id}
```

Prod Goal Line Group

Goals for WMS Activities are captured in the **Prod Goal Line Group** view. The following APIs can be used to get relevant **Prod Goal Line Group** data from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/prod_goal_line_group/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/prod_goal_line_group/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/prod_goal_line_group/{id}
```

Stage Prod Goal Line Group

Goals for WMS activities that are interfaced from external systems using **Line Group Goal** input interface is first written to **Stage Prod Goal Line Group** before validating and moving to **Prod Goal Line Group**. The following APIs can be used to get relevant **Stage Prod Goal Line Group** data.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/stage_prod_goal_line_group/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_prod_goal_line_group/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_prod_goal_line_group/{id}
```

Break

Breaks taken by users are defined in the **Break** view. The following APIs can be used to get relevant **Break** information from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/cw_break/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/cw_break/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/cw_break/{id}
```

Shifts

A facility could be operating in multiple shifts which is defined in the **Shifts** view. The following APIs can be used to get relevant **Shift** information from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/shift/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/shift/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/shift/{id}
```

WFM Item Category

A new view “WFM Item Category” has been introduced to define item categories.

- Item categories will replace SKU Line Groups ONLY for Non-WMS activities
- SKU Line Group will continue to be used for WMS activities

Note: In future, Item Categories will replace SKU Line Groups even for WMS activities. Until then, the following new elements on Items UI can be ignored.

- The “WFM Item Category” field.
- The “UPDATE WFM Item Category” action button.
- The “wfm_item_category” field on input interface.
- Even if WFM Item Category is populated on items, it will not have any functional relevance.

Stage WFM Item Category

WFM Item Category data that is interfaced from external systems using WFM Item Category input interface is first written to Stage WFM Item Category before validating and moving to WFM Item Category. The following APIs can be used to get relevant Stage WFM Item Category data from WFM.

URLs

Check for the existence of the resource:

```
HEAD .../lgfapi/v10/entity/stage_wfm_item_category/{id}
```

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_wfm_item_category/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_wfm_item_category/{id}
```