Oracle Warehouse Management Cloud

WFM REST API Guide

26A

Oracle Warehouse Management Cloud
WFM REST API Guide

26A

G47797-01

Author: Oracle WMS Cloud Product Team

Contents

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 ..lgfapi/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>
 <list-item>
```

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 VAS Activities

The **Prod Goal VAS** API allows you to create one or more goals for VAS (non-WMS) productive activities in WFM.

URL

```
POST ..lgfapi/v10/entity/prod_goal_vas/
```

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)

Request Body Example (JSON)

Instance Create

```
 "fields": {
 "facility_id": 648,
 "work_area_activity_id":494 ,
 "sku_line_group_id": 22,
 "cal_month_id": 6,
 "cal_year": 2023,
 "uom_id": 1,
 "uom_per_hr": 50
 }
}
```

Bulk Create

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

```
 "fields":[
 {
 "facility_id": 648,
 "work_area_activity_id": 513,
 "sku_line_group_id": 22,
 "cal_month_id": 3,
 "cal_year": 2023,
 "uom_id": 1,
 "uom_per_hr": 50
 },
 {
 "facility_id": 648,
 "work_area_activity_id": 513,
 "sku_line_group_id": 22,
 "cal_month_id": 4,
 "cal_year": 2023,
 "uom_id": 1,
 "uom_per_hr": 50
 },
 {
 "facility_id": 648,
 "work_area_activity_id": 513,
 "sku_line_group_id": 22,
 "cal_month_id": 5,
 "cal_year": 2023,
 "uom_id": 1,
 "uom_per_hr": 50
 }
 ]
}
```

Request Body Example (XML)

Instance Create

```
<request>
<fields>
<facility_id>648</facility_id>
<work_area_activity_id>513</work_area_activity_id>
<sku_line_group_id>42</sku_line_group_id>
<cal_month_id>2</cal_month_id>
<cal_year>2023</cal_year>
<uom_id>1</uom_id>
<uom_per_hr>20</uom_per_hr>
</fields>
</request>
```

Bulk Create

```
<request>
 <fields>
 <list-item>
 <facility_id>648</facility_id>
 <work_area_activity_id>553</work_area_activity_id>
 <sku_line_group_id>42</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>
 </list-item>
 <list-item>
 <facility_id>648</facility_id>
 <work_area_activity_id>553</work_area_activity_id>
 <sku_line_group_id>42</sku_line_group_id>
 <cal_month_id>2</cal_month_id>
```

```
<cal_year>2023</cal_year>
<uom_id>1</uom_id>
<uom_per_hr>20</uom_per_hr>
</list-item>
<list-item>
<facility_id>648</facility_id>
<work_area_activity_id>553</work_area_activity_id>
<sku_line_group_id>42</sku_line_group_id>
<cal_month_id>1</cal_month_id>
<cal_year>2023</cal_year>
<uom_id>1</uom_id>
</list-item>
<list-item>
<facility_id>648</facility_id>
<work_area_activity_id>553</work_area_activity_id>
</list-item>
<list-item>
<facility_id>648</facility_id>
<work_area_activity_id>553</work_area_activity_id>
<sku_line_group_id>42</sku_line_group_id>
<cal_month_id>4</cal_month_id>
<cal_year>2023</cal_year>
<uom_id>1</uom_id>
<uom_per_hr>20</uom_per_hr>
</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.

VAS Activity Data

The **Daily VAS Track** API allows you to create one or more VAS activity data records in WFM.

URL

```
POST ..lgfapi/v10/entity/daily_vas_track/
```

Fields

Name	Required	Type	Description
facility_id	X	Integer	Facility ID
work_area_id	X	Integer	WFM Internal ID of the Work Area
work_area_activity_id	X	Integer	WFM Internal ID of the VAS Activity (Work Area Activity of Type 'VAS')

cal_date	X	date	Date in DDMMYYYY format
team_id	O	integer	WFM Internal ID of the team; If this is populated, cw_user_id should not be sent (**Refer Note below)
cw_user_id	O	integer	WFM Internal ID of the user; If this is populated, team_id should not be sent (**Refer Note below)
qty	X	integer	Qty of items handled
uom_id	X	integer	WFM Internal ID of the UOM
Sku_line_group_id	X	integer	WFM Internal ID of the SKU Line Group

Note: To indicate empty values for team_id or cw_user_id, the following conventions should be used:

- In JSON, omit the tag completely or send a value of null.
- In XML, omit the tag completely or send empty tags.

Request Body Example (JSON)

Instance Create

```
{
 "fields": {
 "facility_id": 648,
 "work_area_id”: 516,
 "work_area_activity_id": 476,
 "cal_date": "2024-02-15",
 "team_id": null,
 "cw_user_id" :10754,
 "qty": 150,
 "uom_id": 2,
 "sku_line_group_id": 42
 }
}
```

Bulk Create

```
{
 "options": {
 "commit_frequency": 2
 },
 "fields": [
{
 "facility_id": 648,
 "work_area_id”: 516,
 "work_area_activity_id": 553,
 "cal_date": "2024-02-13",
 "team_id": "",
 "cw_user_id”: "33423391",
 "qty": 120,
 "uom_id": 1,
 "sku_line_group_id": 162
```

```
 },
 {
 "facility_id": 648,
 "work_area_id”: 516,
 "work_area_activity_id": 553,
 "cal_date": "2024-02-13",
 "team_id": 180,
 "cw_user_id”: "",
 "qty": 100,
 "uom_id": 1,
 "sku_line_group_id": 162
 }
 ]
}
```

Request Body Example (XML)

Instance Create

```
<request>
 <fields>
 <facility_id>648</facility_id>
 <work_area_id>1056</work_area_id>
 <work_area_activity_id>893</work_area_activity_id>
 <cal_date>2024-03-08</cal_date>
 <team_id>181</team_id>
 <cw_user_id></cw_user_id>
 <qty>15</qty>
 <uom_id>2</uom_id>
 <sku_line_group_id>42</sku_line_group_id>
 </fields>
```

Bulk Create

```
<request>
 <fields>
 <list-item>
 <facility_id>648</facility_id>
 <work_area_id>1056</work_area_id>
 <work_area_activity_id>476</work_area_activity_id>
 <cal_date>2024-02-13</cal_date>
 <team_id>440</team_id>
 <cw_user_id></cw_user_id>
 <qty>35</qty>
 <uom_id>1</uom_id>
 <sku_line_group_id>2</sku_line_group_id>
 </list-item>
 <list-item>
 <facility_id>648</facility_id>
 <work_area_id>516</work_area_id>
 <work_area_activity_id>478</work_area_activity_id>
 <cal_date>2024-02-13</cal_date>
 <team_id></team_id>
 <cw_user_id>787</cw_user_id>
 <qty>350</qty>
 <uom_id>3</uom_id>
 <sku_line_group_id>249</sku_line_group_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.

Work Area Activities

The **Work Area Activity** API allows you to create one or more System, VAS, or Manual 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 System, VAS or Manual Work Area Activity
name	X	String	Name of the System, VAS or Manual Work Area Activity
work_area_activity_type_id	X	Integer	WFM Internal ID of the type of Work Area Activity. The supported types are System, VAS or Manual Activity
work_area_id	X	integer	WFM Internal ID of Work Area associated to the System, VAS or Manual 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 System Activities

The **WAA Screen XREF** API allows you to associate one or more WMS Screensto System 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 System 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 In/Out

The **Clock** API allows you to create one or more clock records in WFM. Unlike other REST APIs, this API does not post information directly to the main view (Track Record) for clock data. Instead, records are posted to Stage Record tab of Clock in the Input Interfaces view. The scheduled job "Process Stage Track Record" validates and moves these records to Track Record at scheduled intervals. You can choose to trigger the scheduled job immediately as a part of the API request, using the async_flg parameter on the request.

URL

```
POST ..lgfapi/v10/stage_track_record/clock_inout/
```

Request Body

- **message_id** - Mandatory field of string type that would uniquely identify a payload. Max length – 50

- **async_flg** - true/false; default true
  - When false:
    - Posts data to Stage Record of Clock Input Interface and triggers "Process Stage Track Record" immediately after.
    - If all records are processed successfully, response 204 "No Content" is returned with no response body. All the records show up in Track 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 work area clock-in), no records will be processed after the first failure has been encountered.

    > **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 202 - "Accepted" with no response body.
    - Posted records remain in Stage Record of Clock Input Interface for further processing.

- **cico_data** - The table below describes the fields within this parameter:

Name	Required	Type	Description
facility_code	X	String	Facility code
univ_id_1	C	String	Employee ID of the user that needs to be clocked. Mandatory for all cico_types except F7 and F8. If F7 or F8, either univ_id_1 or team should be provided, not both.
cico_ts	X	DateTime	Clock in/out date time Format: YYYYMMDDHHMMSS Example: 20190727052020
cico_type	X	String	Valid Values: F1, F2, F3, F4, F5, F6, F7, F8, F9, F10 F1 - Facility Clock In F2 - Break Clock In F3 - Break Clock Out F4 - Facility Clock Out F5 - Work Area Clock In F6 - Work Area Clock Out

			F7 - VAS Activity Clock In F8 - VAS Activity Clock Out F9 - Manual Activity Clock In F10 - Manual Activity Clock Out
originating_system	X	String	Valid Values: M and S **M** - M is used to create new data. If M is sent for data that already exists, the record fails to be processed **S** - S is used to update existing data. If S is sent for data that doesn't exist, the record is inserted as new data. **Note:** This check is done by "Process Stage Track" process.
work_area	C	String	Work Area Code. Required when cico_type is F5, F6, F7, F8, F9, F10. If F7 or F8, the work area code should be associated to the VAS activity. If F9 or F10, the work area code should be associated to the manual activity.
break	C	String	Name of the break taken by employee. Required when cico_type is F2 or F3.
activities	C	String	Work Area Activity Code of the VAS or Manual activity. Required when cico_type is F7, F8, F9, F10.
team	C		Required only for F7 and F8, if univ_id_1 is not provided. If univ_id_1 is provided, this field should not be populated.

Request Body Example (JSON)

Instance Create

```
{
 "message_id": 12345,
 "async_flg": false,
 "cico_data": [{
 "facility_id__code": "QATST01",
 "univ_id_1": "1115472",
 "cico_ts": "20230505063027",
 "cico_type": "F7",
 "originating_system": "M",
 "work_area": "VASWABN",
 "activities”: "BNVASACT01",
 "team”: ""
 }]
}
```

Bulk Create

```
{
 "message_id": 12345,
 "async_flg": false,
 "cico_data": [
 {
 "facility_id__code": "QATST01",
 "univ_id_1": "",
 "cico_ts": "20230505063027",
 "cico_type": "F7",
 "originating_system": "M",
 "work_area": "VASWABN",
 "activities”: "BNVASACT01",
 "team”: "T345"
 },
 {
 "facility_id__code": "QATST01",
 "univ_id_1": "",
 "cico_ts": "20230505072014",
 "cico_type": "F8",
 "originating_system": "M",
 "work_area": " VASWABN"
 "activities”: " BNVASACT01",
 "team”: "T345"
 }
 ]
}
```

Request Body Example (XML)

Instance Create

```
<request>
 <message_id>12345</message_id>
 <async_flg>true</async_flg>
 <cico_data>
 <list-item>
 <facility_code>QA3PLEST</facility_code>
 <univ_id_1>12453</univ_id_1>
 <cico_ts>20230510092030</cico_ts>
 <cico_type>F9</cico_type>
 <originating_system>M</originating_system>
 <work_area>MISC</work_area>
```

```
 <break></break>
 <activities>Maintenance</activities>
 </list-item>
 </cico_data>
</request>
```

Bulk Create

```
<request>
 <message_id>12345</message_id>
 <async_flg>true</async_flg>
 <cico_data>
 <list-item>
 <facility_code>QA3PLEST</facility_code>
 <univ_id_1>23145</univ_id_1>
 <cico_ts>20230510092030</cico_ts>
 <cico_type>F9</cico_type>
 <originating_system>M</originating_system>
 <work_area>MISC</work_area>
 <break></break>
 <activities>Maintenance</activities>
 </list-item>
 <list-item>
 <facility_code>QA3PLEST</facility_code>
 <univ_id_1>23145</univ_id_1>
 <cico_ts>20230510104511</cico_ts>
 <cico_type>F10</cico_type>
 <originating_system>M</originating_system>
 <work_area>MISC</work_area>
 <break></break>
 <activities>Maintenance</activities>
 </list-item>
 </cico_data>
</request>
```

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	Require	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_coordina	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_coordina	C	Integer	
z_coordina	C	Integer	
Coordina 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_timestan	X	DateTim	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**: o User's default facility time zone: +05:30 o Time zone of the facility_id passed in the payload: +05:30 o Tracked_timestamp sent with any time zone is converted appropriately. **Example 2**: o User's default facility time zone: +08:00 o Time zone of the facility_id passed in the payload: +05:30 o 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

equipme 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
locatic barco	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_coordi	C	Intege	
z_coordi	C	Intege	
coordi 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.
tracke timest	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" o 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
equip 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	Intege	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 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"
 }]
}
```

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}
```

Daily VAS Track

The **Daily VAS Track** view captures VAS (non-WMS) productive activities performed by the users on the warehouse floor. The following APIs can be used to get relevant **Daily VAS Track** data.

URLs

Check for the existence of the resource:

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

Fetch paginated results:

```
GET .../lgfapi/v10/entity/daily_vas_track/
```

Fetch non-paginated result by specific 'ID':

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

Stage Daily VAS Track

VAS (non-WMS) Activities that is interfaced from external systems using the **Daily VAS Track** input interface is first written to **Stage Daily VAS Track** before validating and moving to **Daily VAS Track**. The following APIs can be used to get relevant **Stage Daily VAS Track** data.

URLs

Check for the existence of the resource:

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

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_daily_vas_track/
```

Fetch non-paginated result by specific 'ID':

```
GET …/lgfapi/v10/entity/stage_daily_vas_track/{id}
```

Track Record

The **Track Record** view captures user clock data. The following APIs can be used to get relevant **Track Record** data.

URLs

Check for the existence of the resource:

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

Fetch paginated results:

```
GET .../lgfapi/v10/entity/track_record/
```

Fetch non-paginated result by specific 'ID':

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

Stage Track Record

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

URLs

Check for the existence of the resource:

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

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_track_record/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_track_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 users performing a VAS activity collectively. The following APIs can be used to get relevant **Team** information from WFM.

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 groups of users performing a VAS activity collectively. 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 ssigned 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 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 System (WMS), VAS and Manual 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

System, VAS, and Manual Activities that 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 system activity in WFM. The mapping between WMS RF screens and WFM system 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 System 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 System (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 system 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}
```

Prod Goal VAS

Goals for VAS (non-WMS) productive activities are captured in the **Prod Goal VAS** view. The following APIs can be used to get relevant **Prod Goal VAS** data from WFM.

URLs

Check for the existence of the resource:

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

Fetch paginated results:

```
GET .../lgfapi/v10/entity/prod_goal_vas/
```

Fetch non-paginated result by specific 'ID':

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

Stage Prod Goal VAS

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

URLs

Check for the existence of the resource:

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

Fetch paginated results:

```
GET .../lgfapi/v10/entity/stage_prod_goal_vas/
```

Fetch non-paginated result by specific 'ID':

```
GET .../lgfapi/v10/entity/stage_prod_goal_vas/{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}
```