Contents

Preface .......................................................................................................................... 5
  Audience ....................................................................................................................... 5
  Customer Support ......................................................................................................... 5
  Documentation ............................................................................................................. 5
  Revision History ........................................................................................................... 5

1 Scheduler Web Service ............................................................................................. 6
  Authorization and Authentication ................................................................................. 6
  Create API Users ........................................................................................................... 6
  REST API Versioning .................................................................................................... 7
    Get All API Versions .................................................................................................... 7
    Get Specific API Version ........................................................................................... 9

2 Web Service Calls .................................................................................................... 10
  upload ........................................................................................................................... 10
    URL Param .................................................................................................................. 10
    Sample Request ........................................................................................................ 10
    Sample Response ...................................................................................................... 11
    Success Response ...................................................................................................... 12
    Error Response .......................................................................................................... 12
    Sample Responses ..................................................................................................... 12
  addJob .......................................................................................................................... 13
    URL Param .................................................................................................................. 13
    Sample Request ........................................................................................................ 14
    Sample Response ...................................................................................................... 14
    Success Response ...................................................................................................... 15
    Error Response .......................................................................................................... 16
  editJob .......................................................................................................................... 16
    URL Param .................................................................................................................. 16
    Sample Request ........................................................................................................ 16
    Sample Response ...................................................................................................... 16
    Success Response ...................................................................................................... 18
    Error Response .......................................................................................................... 18
  getJob ........................................................................................................................... 18
    URL Param .................................................................................................................. 18
    Sample Request ........................................................................................................ 19
    Success Response ...................................................................................................... 20
Preface

The Scheduler RESTful API is a Web service interface that allows Labor Management users to add and edit a schedule using a third party labor application, upload the schedule to Labor Management and then enforce the schedule at the point of clock in/out at the POS.

This document specifies the Scheduler RESTful API Web Service interface, its supporting Jobs interfaces, and all available calls.

Audience

This document is intended for developers and programmers creating and configuring third-party applications to interface with Labor Management.

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:
https://support.oracle.com

When contacting Customer Support, please provide the following:
Product version and program/module name
Functional and technical description of the problem (include business impact)
Detailed step-by-step instructions to re-create
Exact error message received and any associated log files
Screen shots of each step you take

Documentation

Product documentation is available on the Oracle Help Center at https://docs.oracle.com/en/industries/food-beverage/

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 2017</td>
<td>Initial publication</td>
</tr>
<tr>
<td>March 2018</td>
<td>Changed the upload response template to include the failedSchedules response, and added sample responses.</td>
</tr>
<tr>
<td>September 2018</td>
<td>• Updated REST HTTP header and security information</td>
</tr>
<tr>
<td></td>
<td>• Added information of REST API versioning</td>
</tr>
<tr>
<td>October 2018</td>
<td>Added Authentication and Authorization section.</td>
</tr>
<tr>
<td>May 2019</td>
<td>Added supported Enterprise Back Office version information.</td>
</tr>
</tbody>
</table>
1 Scheduler Web Service

Authorization and Authentication

When calling a web service, you pass the following parameters in the HTTP header:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
<td>Application/json</td>
</tr>
<tr>
<td>Username</td>
<td>CORE_ORGANIZATION.externa1AppUserName</td>
</tr>
<tr>
<td>Password</td>
<td>CORE_ORGANIZATION.externa1AppPassword</td>
</tr>
</tbody>
</table>

Starting with Enterprise Back Office 9.0 patch 17 and higher and 9.1 patch 4 and higher, if you are developing new REST APIs that follow Oracle REST standards, use an API token and password in the HTTP request header to authenticate client requests.

Create API Users explains how to create an API user to get the user name token and password.

The following table describes the header parameters for the new APIs:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>API user token associated with the API user.</td>
</tr>
<tr>
<td>password</td>
<td>Password associated with the API user.</td>
</tr>
<tr>
<td>Content-Type</td>
<td>application/json</td>
</tr>
</tbody>
</table>

Create API Users

Authentication tokens for the API calls are generated when you create and save an API user. When an API call is made, the API uses the token along with a password that are included in the connection string to authenticate and authorize users making the calls.

This combination of token and password also validates which locations the user is authorized to update.

1. In the portal, select Admin, select Portal, select Users, and then select Users.
2. Click Add.
3. Select API User and enter details for all mandatory fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logon Name</td>
<td>The username that the user should use to log in to the portal.</td>
</tr>
<tr>
<td>Password</td>
<td>The password that the user should use to log in to the portal.</td>
</tr>
<tr>
<td>Display Name</td>
<td>The name that should show appear in the portal when the user is logged in.</td>
</tr>
<tr>
<td>Organization Level</td>
<td>The user’s organization levels.</td>
</tr>
<tr>
<td>Email Address</td>
<td>The user’s email address.</td>
</tr>
</tbody>
</table>
Active

Enabled by default. Disable this option if you want to deactivate the user.

4. Click **Save**.
5. If you want to view a list of all API users, select **Show API Users** and click **Find All** in the User Administration overview page.

**REST API Versioning**

The new implementation of the Labor Management REST APIs follows Oracle REST standards. REST API calls include the API version in the root URL. Root URLs are formatted as follows:

```
https://www.hostname.com/rest/services/{version}/{api}
```

The path to the resource appears after the root URL. The version is highlighted in the following example API call:

```
https://www.hostname.com/rest/services/v1/locations/{locationRef}/employees
```

When the next version of the API is released, you can reference the new version in the call as follows:

```
https://www.hostname.com/rest/services/v2/locations/{locationRef}/employees
```

**Get All API Versions**

Use the GET method with the following URL to retrieve all versions of an API:

```
https://www.hostname.com/rest/services/
```
Sample Response
The following example shows the information returned in the JSON response:

```json
{
  "count": 2,
  "items": [
    {
      "version": "v1",
      "lifecycle": "deprecated",
      "isLatest": false,
      "successor": "v2",
      "terminationDate": "2017-12-31",
      "links": [
        {"rel": "canonical",
        "href": "HTTPS://machine-name/rest/services/v1"},
        {"rel": "successor-version",
        "href": "HTTPS://machine-name/rest/services/v2"}
      ]
    },
    {
      "version": "v2",
      "lifecycle": "active",
      "isLatest": true,
      "predecessor": "v1",
      "links": [
        {"rel": "canonical",
        "href": "HTTPS://machine-name/rest/services/v2"},
        {"rel": "predecessor-version",
        "href": "HTTPS://machine-name/rest/services/v1"}
      ]
    }
  ],
  "links": [
    {"rel": "canonical",
    "href": "HTTPS://machine-name/rest/services"},
    {"rel": "current",
    "href": "HTTPS://machine-name/rest/services/v2"}
  ]
}
```
Get Specific API Version

Use the GET method with the following URL to retrieve information for a specific API version:
https://www.hostname.com/rest/services/{version}

URL Parameters

The following table describes the URL parameters for retrieving a specific version of an API:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>Version of the API.</td>
</tr>
</tbody>
</table>

Sample Request

https://www.hostname.com/rest/services/v1

Sample Response

The following example shows the information returned in the JSON response:

```
{
    "version": "v1",
    "lifecycle": "deprecated",
    "isLatest": false,
    "successor": "v2",
    "terminationDate": "2017-12-31",
    "links": [
        {"rel": "canonical",
         "href": "HTTPS://machine-name/rest/services/v1"},
        {"rel": "successor-version",
         "href": "HTTPS://machine-name/rest/services/v2"}
    ]
}
```
2 Web Service Calls

The Oracle Hospitality Labor Management Scheduler API lets you:

- Import a schedule from a third-party application.
- Create and edit Job Definitions.
- Retrieve information pertaining to a specific Job Definition or for all Job Definitions for an organization.

The following table describes the calls:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>upload</td>
<td>Upload and import a schedule to Labor Management.</td>
</tr>
<tr>
<td>addJob</td>
<td>Create a Job Definition.</td>
</tr>
<tr>
<td>editJob</td>
<td>Edit an existing Job Definition.</td>
</tr>
<tr>
<td>getJob</td>
<td>Retrieve a specific Job Definition.</td>
</tr>
<tr>
<td>getJobs</td>
<td>Retrieve all Job Definitions for an organization.</td>
</tr>
</tbody>
</table>

When making a web service call, make sure there is a value in the Email Address field for the Schedule Profile, and enable Disable Labor Cost Calculation (Scheduler API). The e-mail address is used to send notifications if there is a problem in uploading a schedule.

The user can specify multiple e-mail addresses with a semi-colon. For example: user1@test.com;user2@test.com;user3@test.com.

**upload**

This is used to upload the schedule to Labor Management.

**URL Param**

http://server name/rest/services/schedule/upload

New URL with API versioning (supported in 9.0 patch 17 and higher and 9.1 patch 4 and higher):

http://server name/rest/services/v1/schedules/upload

**Sample Request**

```
upload ({{"locationRef": "<val>"}, "sched": [{
    "payrollId": "<val>", "startDate": "<val>",
    "startHour": "<val>", "startInterval": "<val>",
    "endDate": "<val>", "endHour": "<val>",
    "endInterval": "<val>",
    "jobRef": "<val>"
}}
```

The failedSchedules response is only populated when returning error code 1. The response includes the following parameters and values. Labor Management cannot import a schedule if any of the following parameters are missing:

**Table 3 - upload Parameter Reference**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>locationRef</td>
<td>The Location Reference as defined in the column ORGLOCATIONREF. It can be found at Admin</td>
</tr>
<tr>
<td>payrollId</td>
<td>The Employees Eternal Payroll ID as defined in column EXTERNALPAYROLLID. It can be found at Human Resources</td>
</tr>
<tr>
<td>startDate</td>
<td>The Shift Start Date as defined in column SCHEDULEDATE. The date field should be in the MM/dd/yyyy format, such as 01/26/2015.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>startHour</td>
<td>The Shift Start Hour as defined in column STARTFP, consisting of StartHour + StartInterval.</td>
</tr>
<tr>
<td>startInterval</td>
<td>The Shift Start Interval as defined in column STARTFP, consisting of StartHour + StartInterval.</td>
</tr>
<tr>
<td>endDate</td>
<td>The Shift End Date, expected to be the same value as StartDate. There is no equivalent column in the database. The date field should be in the MM/dd/yyyy format, such as 01/26/2015.</td>
</tr>
<tr>
<td>endHour</td>
<td>The Shift End Hour as defined in column ENDFP, consisting of EndHour + EndInterval.</td>
</tr>
<tr>
<td>endInterval</td>
<td>The Shift End Interval as defined in column ENDFP, consisting of EndHour + EndInterval.</td>
</tr>
<tr>
<td>jobRef</td>
<td>The Job Code assigned to the Employee record as defined in column EXT_JOB_CODE.EXTJOBCODEREF.</td>
</tr>
</tbody>
</table>

**Success Response**

0. Success – Complete Schedule upload

**Error Response**

1. Partial Success - Some schedules failed validation.
2. Server Error
3. Invalid Credentials
5. Configuration Error – No schedule profile or no organization found.

**Sample Responses**

**Successful Upload**

```
{
   "response": {
      "code": 0,
      "desc": "Complete Success. All schedules were saved."
   }
}
```

**Upload Containing Invalid Data**

```
{
   "response": {
      "code": 1,
      "desc": "Partial Success. Some schedules were not saved."
   }
}
```

```
   "failedSchedules": {
      "locationRef": "BAL",
      "failures": [  
```
addJob

This is used to add (create) a Job Definition to Labor Management. When creating a new Job Definition, the following elements are required:

- Job
- JobCatPosRef
- LabCatPosRef
- RvcPosRef
- Type
- EffectiveFrom

URL Param

http://server name/rest/services/job/addJob

New URL with API versioning (supported in 9.0 patch 17 and higher and 9.1 patch 4 and higher):

http://server name/rest/services/v1/jobs/addJob

REST API Versioning provides more information on API versions.
Sample Request

```json
addJob({
    "job": {
"name": "<val>", "jobcatposref": "<val>",
"labcatposref": "<val>", "rvcposref": "<val>",
"type": "<val>", "effectivefrom": "<val>",
"clockinprivilege": "<val>",
"declaredirecttipsonclockout": "<val>",
"declareindirecttipsonclockout": "<val>",
"allowearlylateclockout": "<val>", "printtips": "<val>",
"tipped": "<val>", "jobRef": "<val>", "defaultlate": "<val>",
"extjobcodeid": "<val>", "override": [{
    "locationRef": "<val>",
    "effectivefrom": "<val>",
    "defaultlate": "<val>
    }
  ]
})
})
```

Sample Response

```json
{
"response": {
"code": "<val>",
"desc": "<val>"
}
}
```

The response includes the following parameters and values.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The Job Name defined in [Human Resources</td>
</tr>
<tr>
<td>jobcatposref</td>
<td>The Job Category defined in [Human Resources</td>
</tr>
<tr>
<td>labcatposref</td>
<td>The Labor Category defined in [Human Resources</td>
</tr>
<tr>
<td>rvcposref</td>
<td>The Revenue Center in which the job is effective, as defined in [Human Resources</td>
</tr>
<tr>
<td>type</td>
<td>The Job Type, such as Hourly or Salaried, as defined in [Human Resources</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>effectivefrom</td>
<td>The date the job is Effective From as defined in Human Resources</td>
</tr>
<tr>
<td>clockinprivilege</td>
<td>The privilege that an Employee must have in order to clock-in for this position, as defined in Human Resources</td>
</tr>
<tr>
<td>declaredirecttips</td>
<td>Whether or not the job requires a declaration of direct tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>declareindirecttips</td>
<td>Whether or not the job requires a declaration of indirect tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>allowearlylateclockout</td>
<td>Whether or not employees working this job can clock out early or late, even when their Employee class is not privileged to do so. The configuration is defined in Human Resources</td>
</tr>
<tr>
<td>printtips</td>
<td>Whether or not to print a tips declared chit when an Employee working this job clocks out, as defined in Human Resources</td>
</tr>
<tr>
<td>tipped</td>
<td>Whether or not the job is subject to calculating the tip credit, as defined in Human Resources</td>
</tr>
<tr>
<td>Jobref</td>
<td>The Job Code assigned to the Employee record.</td>
</tr>
<tr>
<td>defaultrate</td>
<td>The normal starting pay for this position, as defined in Human Resources</td>
</tr>
<tr>
<td>locationRef</td>
<td>The Reference ID associated with the Location or Level, as defined in Admin</td>
</tr>
<tr>
<td>extjobcodeid</td>
<td>The external job reference for the newly added job. This is stored in the ENT_JOB_CODE.extJobCodeRef column.</td>
</tr>
</tbody>
</table>

**Success Response**

0. Success
**Error Response**

2. Server Error  
3. Invalid Credentials  
4. Malformed Data  
5. Configuration Error  

**editJob**

This is used to edit a Job Definition in Labor Management.

**URL Param**

http://server name/rest/services/job/editJob

New URL with API versioning (supported in 9.0 patch 17 and higher and 9.1 patch 4 and higher):

http://server name/rest/services/schedule/v1/jobs/editJob

REST API Versioning provides more information on API versions.

**Sample Request**

```
editJob(String oldJobRef, {"job": {
    "name":"<val>", "jobcatposref":"<val>",
    "labcatposref":"<val>", "rvcposref":"<val>",
    "effectivefrom":"<val>", "clockinprivilege":"<val>",
    "declaredirecttipsonclockout":"<val>",
    "declareindirecttipsonclockout":"<val>",
    "allowearlylateclockout":"<val>",
    "printtips":"<val>", "tipped":"<val>",
    "jobRef":"<val>", "defaultrate":"<val>",
    "extjobcodeid":"<val>", "override": [{
        "locationRef":"<val>",
        "effectivefrom":"<val>",
        "defaultrate":"<val>
    }]
}})
```

**Sample Response**

```
{"response": {"code":"<val>",
    "desc":"<val>"}
}
```
The response includes the following parameters and values.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The Job Name defined in Human Resources</td>
</tr>
<tr>
<td>jobcatposref</td>
<td>The Job Category defined in Human Resources</td>
</tr>
<tr>
<td>labcatposref</td>
<td>The Labor Category defined in Human Resources</td>
</tr>
<tr>
<td>rvcposref</td>
<td>The Revenue Center in which the job is effective, as defined in Human Resources</td>
</tr>
<tr>
<td>type</td>
<td>The Job Type, such as Hourly or Salaried, as defined in Human Resources</td>
</tr>
<tr>
<td>effectivefrom</td>
<td>The date the job is Effective From as defined in Human Resources</td>
</tr>
<tr>
<td>clockinprivilege</td>
<td>The privilege that an Employee must have in order to clock-in for this position, as defined in Human Resources</td>
</tr>
<tr>
<td>declaredirecttips</td>
<td>Whether or not the job requires a declaration of direct tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>onclockout</td>
<td></td>
</tr>
<tr>
<td>declareindirecttips</td>
<td>Whether or not the job requires a declaration of indirect tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>onclockout</td>
<td></td>
</tr>
<tr>
<td>allowearlylateclockout</td>
<td>Whether or not employees working this job can clock out early or late, even when their Employee class is not privileged to do so. The configuration is defined in Human Resources</td>
</tr>
<tr>
<td>printtips</td>
<td>Whether or not to print a tips declared chit when an Employee working this job clocks out, as defined in Human Resources</td>
</tr>
</tbody>
</table>
tipped
Whether or not the job is subject to calculating the tip credit, as defined in Human Resources | Admin | Job Definitions | General | Subject to Tipped Credit.

Jobref
The Job Code assigned to the Employee record.

defaultrate
The normal starting pay for this position, as defined in Human Resources | Admin | Job Definitions | Type: Hourly | Default Pay Rate. This configuration is only required for hourly employees.

locationRef
The Reference ID associated with the Location or Level, as defined in Admin | Portal | Organizational Structure | Locations | Location Reference.

extjobcodeid
The external job reference for the newly added job. This is stored in the ENT_JOB_CODE.extJobCodeRef column.

Success Response
0. Success

Error Response
2. Server Error
3. Invalid Credentials
4. Malformed Data
5. Configuration Error

getJob
This is used to retrieve a single Job Definition from Labor Management.

URL Param
http://server name/rest/services/job/getjob

New URL with API versioning (supported in 9.0 patch 17 and higher and 9.1 patch 4 and higher):
http://server name/rest/services/v1/jobs/getJob

REST API Versioning provides more information on API versions.
Sample Request

getJob (String oldJobRef)

The API returns the following response:-

```
{"job":{
        "locationRef"":"<val>",
        "effectiveFrom"":"<val>",
        "defaultrate"":"<val>
    }
    ]
},
"response":{
    "code"":"<val>",
    "desc"":"<val>
}
```

The response includes the following parameters and values.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The Job Name defined in **Human Resources</td>
</tr>
<tr>
<td>jobcatposref</td>
<td>The Job Category defined in **Human Resources</td>
</tr>
<tr>
<td>labcatposref</td>
<td>The Labor Category defined in **Human Resources</td>
</tr>
<tr>
<td>rvcposref</td>
<td>The Revenue Center in which the job is effective, as defined in **Human Resources</td>
</tr>
<tr>
<td>type</td>
<td>The Job Type, such as Hourly or Salaried, as defined in **Human Resources</td>
</tr>
<tr>
<td>effectivefrom</td>
<td>The date the job is Effective From as defined in **Human Resources</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>clockinprivilege</td>
<td>The privilege that an Employee must have in order to clock-in for this position, as defined in Human Resources</td>
</tr>
<tr>
<td>declaredirecttips</td>
<td>Whether or not the job requires a declaration of direct tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>declareindirecttips</td>
<td>Whether or not the job requires a declaration of indirect tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>allowearlylateclockout</td>
<td>Whether or not employees working this job can clock out early or late, even when their Employee class is not privileged to do so. The configuration is defined in Human Resources</td>
</tr>
<tr>
<td>printtips</td>
<td>Whether or not to print a tips declared chit when an Employee working this job clocks out, as defined in Human Resources</td>
</tr>
<tr>
<td>tipped</td>
<td>Whether or not the job is subject to calculating the tip credit, as defined in Human Resources</td>
</tr>
<tr>
<td>defaultrate</td>
<td>The normal starting pay for this position, as defined in Human Resources</td>
</tr>
<tr>
<td>locationRef</td>
<td>The Reference ID associated with the Location or Level, as defined in Admin</td>
</tr>
<tr>
<td>extjobcodeid</td>
<td>The external job reference for the newly added job. This is stored in the ENT_JOB_CODE.extJobCodeRef column.</td>
</tr>
</tbody>
</table>

**Success Response**

0. Success

Web Service Calls
Error Response
2. Server Error
3. Invalid Credentials
4. Malformed Data
5. Configuration Error

getJobs
This is used to retrieve all the Job Definitions of the Organization from Labor Management.

URL Param
http://server name/rest/services/job/getJobs

New URL with API versioning (supported in 9.0 patch 17 and higher and 9.1 patch 4 and higher):
http://server name/rest/services/v1/jobs/getJobs

REST API Versioning provides more information on API versions.

Sample Request
getJobs ()

Sample Response
{"jobs": [{
   "name": "<val>", "jobcatposref": "<val>",
   "labcatposref": "<val>", "rvcposref": "<val>",
   "type": "<val>", "effectiveFrom": "<val>",
   "clockinprivilege": "<val>",
   "declaredirecttipsonclockout": "<val>",
   "declareindirecttipsonclockout": "<val>",
   "allowearlylateclockout": "<val>",
   "printtips": "<val>", "tipped": "<val>",
   "jobRef": "<val>", "defaultrate": "<val>",
   "extjobcodeid": "<val>", "override": [{
      "locationRef": "<val>",
      "effectiveFrom": "<val>",
      "defaultrate": "<val>"
   }]
}]
}
The response includes the following parameters and values.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The Job Name defined in Human Resources</td>
</tr>
<tr>
<td>jobcatposref</td>
<td>The Job Category defined in Human Resources</td>
</tr>
<tr>
<td>labcatposref</td>
<td>The Labor Category defined in Human Resources</td>
</tr>
<tr>
<td>rvcposref</td>
<td>The Revenue Center in which the job is effective, as defined in Human Resources</td>
</tr>
<tr>
<td>type</td>
<td>The Job Type, such as Hourly or Salaried, as defined in Human Resources</td>
</tr>
<tr>
<td>effectivefrom</td>
<td>The date the job is Effective From as defined in Human Resources</td>
</tr>
<tr>
<td>clockinprivilege</td>
<td>The privilege that an Employee must have in order to clock-in for this position, as defined in Human Resources</td>
</tr>
<tr>
<td>declaredirecttips onclockout</td>
<td>Whether or not the job requires a declaration of direct tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>declareindirecttips onclockout</td>
<td>Whether or not the job requires a declaration of indirect tips on clock out, as defined in Human Resources</td>
</tr>
<tr>
<td>allowearlylateclockout</td>
<td>Whether or not employees working this job can clock out early or late, even when their Employee class is not privileged to do so. The configuration is defined in Human Resources</td>
</tr>
</tbody>
</table>
printtips Whether or not to print a tips declared chit when an Employee working this job clocks out, as defined in Human Resources | Admin | Job Definitions | General | Print tips declared on clock out.

tipped Whether or not the job is subject to calculating the tip credit, as defined in Human Resources | Admin | Job Definitions | General | Subject to Tipped Credit.

Jobref The Job Code assigned to the Employee record.

defaultrate The normal starting pay for this position, as defined in Human Resources | Admin | Job Definitions | Type: Hourly | Default Pay Rate. This configuration is only required for hourly employees.

locationRef The Reference ID associated with the Location or Level, as defined in Admin | Portal | Organizational Structure | Locations | Location Reference.

extjobcodeid The external job reference for the newly added job. This is stored in the ENT_JOB_CODE.extJobCodeRef column.

Success Response

0. Success

Error Response

2. Server Error
3. Invalid Credentials
4. Malformed Data
5. Configuration Error