

List of API's & Screenshots:

>> Created Get Employee API along with skill id using association

Method: POST

Request: Login user name

Based on the user name, it will fetch all the record

The screenshot displays a REST client interface for a project named "My Demo / Employee Skill". The request is a POST to the endpoint `http://localhost:8000/api/employeeswithskill`. The request body is a JSON object with a single field: `"username": "lokesk"`. The response is a 200 OK status with a response time of 47 ms and a size of 2.63 KB. The response body is a JSON object containing employee details and a list of skill maps.

Request:

```
POST http://localhost:8000/api/employeeswithskill
```

Request Body (JSON):

```
{  "username": "lokesk"}
```

Response (JSON):

```
{  "employee_id": 1000,  "name": "Ram Krishna",  "status": "bench",  "manager": "rohit",  "wfm_manager": "lokesk",  "email": "writevinodh@gmail.com",  "lockstatus": "not_requested",  "experience": "7",  "skillmaps": [    {      "employee_id": 1000,      "skills": {        "name": "Angular"      }    }  ]}
```

>> Add Soft Lock:

Method: POST

Request: {
 "empld": "1017",
 "manager": "rohit",
 "reqMsg": "Test123"
}

New data will insert into Softskill Table.

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8000/api/addsoftlock ...
- Body:** A JSON object with the following fields:

```
{  "empld": "1017",  "manager": "rohit",  "reqMsg": "Test123"}
```
- Response:** 200 OK, 38 ms, 272 B. The response body is "Data inserted".
- Buttons:** Send, Beautify, Save Response, Runner.

>> Update Soft Lock:

Method: PUT

Request: {
 "reqMsg": "Test message 123",
 "lockID": 1018
}

Request Message will get update into Softskill Table.

The screenshot shows a REST client interface with the following details:

- Method:** POST (dropdown menu)
- URL:** http://localhost:8000/api/updatesoftlock
- Send Button:** A blue button with a dropdown arrow.
- Tabs:** Params, Authorization, Headers (8), **Body** (selected), Pre-request Script, Tests, Settings.
- Body Type:** none, form-data, x-www-form-urlencoded, **raw** (selected), binary, GraphQL, JSON (dropdown).
- Body Content:**

```
1 {  
2   "reqMsg": "Test message 123",  
3   "lockID": 1018  
4 }
```
- Response Section:**
 - Tabs:** **Body** (selected), Cookies, Headers (8), Test Results.
 - Status:** 200 OK, 109 ms, 288 B.
 - Action:** Save Response (dropdown).
 - View Options:** Pretty (selected), Raw, Preview, Visualize, HTML (dropdown), and a refresh icon.
 - Response Content:**

```
1 Request lock message updated
```

>> Update Employee Table with soft lock status as "Waiting":

Method: PUT

Request: {
 "reqMsg": "Test message 123",
 "lockID": 1018
}

Request Message will get update into Softskill Table.

The screenshot shows a REST client interface with the following details:

- URL:** http://localhost:8000/api/updateEmployee
- Method:** POST (dropdown menu)
- Body:**

```
{  "empID": "1023"}
```
- Response:** 200 OK, 155 ms, 288 B. The response body is:

```
Employee lock status updated
```

WFM Manager Home:

Work Force Management						
Emp ID	Name	Skills			Experience	Manager
1000	Ram Krishna	Spring Boot	Core Java	Angular	7	rohit
1001	Janardhan	PL SQL	Core Java	SQL	4	rohit
1002	Deeraj Keshav	React JS	Core Java		5	rohit
1003	Abdul Kadar	Python	SQL		2	rohit
1004	John Paul	React	Python		7	rohit
1017	Gangaram	Python	SQL		8	karan
1018	Shreya	React	Python	SQL	2	karan
1019	Ashwini Kulkarni	React	Core Java		2	karan
1020	Amulya	SQL	Spring Boot	Angular	7	karan
1021	Vidharth	PL SQL	Core Java	SQL	2	karan

WFM Manager SoftLock Add Request:

The screenshot shows a web browser at localhost:3000/home. A modal dialog titled "Soft lock Request Confirmation" is open. It contains a text input field with the placeholder "Request message(message must be atleast 0 char long)". Below the input field are two buttons: "Cancel" and "Send Request". The background shows a table of employees with columns for Emp ID, Name, Skills, and Manager. The table lists 11 employees, each with a "Request Lock" button next to their manager's name.

Emp ID	Name	Skills	Manager
1000	Ram Krishna	Spring	rohit
1001	Janardhan	PL S	rohit
1002	Deeraj Keshav	React	rohit
1003	Abdul Kadar	Python	rohit
1004	John Paul	React, Python	rohit
1017	Gangaram	Python, SQL	karan
1018	Shreya	React, Python, SQL	karan
1019	Ashwini Kulkarni	React, Core Java	karan
1020	Amulya	SQL, Spring Boot, Angular	karan
1021	Vidharth	Python, SQL	karan

Manager Home:

The screenshot shows the "Manager Home" page in a web browser at localhost:3000/home. The page has a blue header with the text "Manager Home" and a user profile icon for "rohit". Below the header is a table with columns for Emp ID, Requestee, and Employee Manager. Each row has a "View Details" button next to the Employee Manager name.

Emp ID	Requestee	Employee Manager
1000	Ram Krishna	rohit
1001	Janardhan	rohit
1002	Deeraj Keshav	rohit
1003	Abdul Kadar	rohit
1004	John Paul	rohit
1017	Gangaram	karan
1018	Shreya	karan
1019	Ashwini Kulkarni	karan
1020	Amulya	karan
1021	Vidharth	karan

Manager Soft Lock Approve Popup:

