

# Meeting Calendar Assistant APIs

## 1. Register Employee

API Endpoint: `/api/v1/employee/register?employeeName=<value>`

Method Type: POST

Output:

- Http Status Code 201 [In case of success] + Employee ID [In Body]
- Http Status Code 400 [In case of failure / exception]

Sample Output (Postman):

The screenshot shows a Postman interface for a POST request to `http://localhost:8080/api/v1/employee/register?employeeName=Shivam Gupta`. The request is in the 'Params' tab, showing a query parameter `employeeName` with the value `Shivam Gupta`. The response is in the 'Body' tab, showing a JSON object `{"employeeId": 1}` with a status of `201 Created`, time of `605 ms`, and size of `274 B`.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> employeeName	Shivam Gupta	
Key	Value	Description

```
1 {
2   "employeeId": 1
3 }
```

The screenshot shows a SQL query editor with the query `select * from employee_details;` and a result grid displaying the data. The result grid has columns `employee_id` and `employee_name`, with one row showing `1` and `Shivam Gupta`.

```
1 • select * from employee_details;
```

employee_id	employee_name
1	Shivam Gupta

## 2. Book Meeting

API Endpoint: /api/v1/meeting/book

Method Type: POST

### Sample Request Body:

```
{
  "title": "Sprint Planning",
  "date": "2021-06-28",
  "schedule": {
    "startTime": {
      "hour": 11,
      "minute": 30
    },
    "endTime": {
      "hour": 13,
      "minute": 30
    }
  },
  "employeeId": [1, 2, 3]
}
```

### Sample Response: [Http Status Code: 201]

```
{
  "meetingId": 4
}
```

The screenshot displays a REST client interface with a POST request to `http://localhost:8080/api/v1/meeting/book`. The request body is a JSON object with the following structure:















```
{
  "title": "Sprint Planning",
  "date": "2021-06-28",
  "schedule": {
    "startTime": {
      "hour": 11,
      "minute": 30
    },
    "endTime": {
      "hour": 13,
      "minute": 30
    }
  },
  "employeeId": [1, 2, 3]
}
```

The response is a JSON object with a status code of 201 (Created) and a response body:

```
{
  "meetingId": 4
}
```








The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests, and Settings. The Body tab is selected, and the request body is shown in a code editor. The response is also shown in a code editor, and the status code is 201. The interface also includes a 'Send' button, a 'Save' button, and a 'Beautify' button.

Query 1 ×















         Limit to 1000 rows ▼     

```
1 • select * from employee_details;
```

<








Result Grid  Filter Rows:  Edit:    Export/Import:   Wrap Cell Content: 

	employee_id	employee_name
▶	1	Shivam Gupta
	2	Rahul Saini
	3	Ramesh Singh
*	NULL	NULL















         Limit to 1000 rows ▼     

```
1 • select * from meeting_details;  
2
```

<








Result Grid  Filter Rows:  Edit:    Export/Import:   Wrap Cell Content: 

	meeting_id	date	end_time	meeting_title	start_time
▶	4	2021-06-28	8:10	Sprint Planning	6:90
*	NULL	NULL	NULL	NULL	NULL

         Limit to 1000 rows ▼     

```
1 select * from meeting_participants
```

<

Result Grid  Filter Rows:  Edit:    Export/Import:   Wrap Cell Content: 

	serial_number	employee_id	meeting_id
▶	5	1	4
	6	2	4
	7	3	4
*	NULL	NULL	NULL

### 3. Find Free Slots for Meeting

API Endpoint: /api/v1/meeting/available

Method Type: GET

#### Sample Request Body:

```
{
  "employeeCalendar": [
    [
      {
        "startTime": {
          "hour": 9,
          "minute": 0
        },
        "endTime": {
          "hour": 9,
          "minute": 30
        }
      },
      {
        "startTime": {
          "hour": 10,
          "minute": 0
        },
        "endTime": {
          "hour": 10,
          "minute": 30
        }
      }
    ],
    [
      {
        "startTime": {
          "hour": 9,
          "minute": 0
        },
        "endTime": {
          "hour": 9,
          "minute": 45
        }
      },
      {
        "startTime": {
          "hour": 10,
          "minute": 15
        },
        "endTime": {
          "hour": 11,
          "minute": 10
        }
      }
    ]
  ]
}
```

```

    }
  ],
  "meetingDuration": 45
}

```

### Sample Response:

```

{
  "availableSlots": [
    {
      "startTime": {
        "hour": 0,
        "minute": 0
      },
      "endTime": {
        "hour": 9,
        "minute": 0
      }
    },
    {
      "startTime": {
        "hour": 11,
        "minute": 10
      },
      "endTime": {
        "hour": 23,
        "minute": 59
      }
    }
  ]
}

```

GET
http://localhost:8080/api/v1/meeting/available
Send
Save

Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Code

none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify

```

1 {
2   "employeeCalendar": [
3     [
4       {
5         "startTime": {
6           "hour": 9,
7           "minute": 0
8         },
9         "endTime": {
10          "hour": 9,
11          "minute": 30
12        }
13      },
14      {
15        "startTime": {
16          "hour": 10,
17          "minute": 0
18        },
19        "endTime": {
20          "hour": 10,
21          "minute": 30

```

Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 84 ms
Size: 413 B
Save Response

#### 4. Find Meeting Conflicts:

API Endpoint: /api/v1/meeting/conflict

Method Type: GET

##### Sample Request:

```
{
  "employees": [
    {
      "employeeId": 1,
      "employeeName": "Shivam Gupta",
      "calendar": [
        {
          "startTime": {
            "hour": 9,
            "minute": 0
          },
          "endTime": {
            "hour": 9,
            "minute": 30
          }
        },
        {
          "startTime": {
            "hour": 10,
            "minute": 0
          },
          "endTime": {
            "hour": 11,
            "minute": 45
          }
        }
      ]
    },
    {
      "employeeId": 2,
      "employeeName": "Rahul Saini",
      "calendar": [
        {
          "startTime": {
            "hour": 9,
            "minute": 0
          },
          "endTime": {
            "hour": 10,
            "minute": 0
          }
        },
        {
          "startTime": {
```

```

        "hour": 11,
        "minute": 30
      },
      "endTime": {
        "hour": 12,
        "minute": 0
      }
    }
  ]
},
{
  "employeeId": 3,
  "employeeName": "Ramesh Singh",
  "calendar": [
    {
      "startTime": {
        "hour": 8,
        "minute": 30
      },
      "endTime": {
        "hour": 9,
        "minute": 0
      }
    },
    {
      "startTime": {
        "hour": 9,
        "minute": 0
      },
      "endTime": {
        "hour": 12,
        "minute": 30
      }
    }
  ]
}
],
"meetingRequestDetail": {
  "startTime": {
    "hour": 10,
    "minute": 30
  },
  "endTime": {
    "hour": 11,
    "minute": 0
  }
}
}

```

### Sample Response:

```
{
  "employeeId": [
    1,
    3
  ]
}
```

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

- Request Bar:** Method: GET, URL: `http://localhost:8080/api/v1/meeting/conflict`. Buttons: Send, Save.
- Request Body:** Tab: Body. Format: JSON. Content: `{,}`.
- Response Section:** Status: 200 OK, Time: 22 ms, Size: 273 B. Buttons: Save Response.
- Response Body:** Tab: Pretty. Format: JSON. Content:

```
1 {
2   "employeeId": [
3     1,
4     3
5   ]
6 }
```