

# API Specification Doc

*(Employee Portal)*

Version	Date	Author	Description
1.0	03-29-2018	Ashish Balani	Initial draft

# Index

## 1. login

[Request](#)

[Response](#)

## 2. register

[Request](#)

[Response](#)

## 3. employees

[Request](#)

[Response](#)

## 4. employee

[Request](#)

[Response](#)

[Conventions](#)

[Status Codes](#)

# Methods

## 1. login

Authenticate the user with the system

### Request

Method	URL
POST	api/users/login/

Type	Params	Values
POST	map	Map<String , String>

map

Map contains username and password to authenticate the user

### Response

Status	Response
200	<b>Provides the user that is authenticated</b> Example: { "id": 5, "email": "ashish.balani.94@gmail.com", "password": "Ashish@94", "fname": "Ashish", "lname": "Balani", "empty": false }
401	{"error": "Incorrect username or password."}
500	{"error": "Something went wrong. Please try again later."}

## 2. register

Add the user to database

### Request

Method	URL
POST	api/user/register

Type	Params	Values
POST	user	User

### user

The model of a user, which contains all the information encapsulated in it, ie. key,value pairs

### Response

Status	Response
200	<p>Response will be an object containing the User added .User has the following structure.</p> <pre>{     "fname": String,     "lname": String,     "email": String,     "address": String,     "phone": String,     "hireDate": Date,     "position": String,     "salary": Double,     "department": String,     "id": int }</pre>

	<b>An example response is:-</b> <pre> {     "fname": "Rakesh",     "lname": "Mahodya",     "email": "rakesh.maho@gmail.com",     "address": "88 Churchil Street",     "phone": "3372237878",     "hireDate": "2018-01-09",     "position": "Software Engineer",     "salary": 70000,     "department": "Development",     "id": 2 } </pre>
400	<pre>{"error": "User not found"}</pre>
500	<pre>{"error": "Something went wrong. Please try again later."}</pre>

### 3. employees

Gets a List of employees in normal order or you can specify the order

#### Request

Method	URL
POST GET	api/employee/ api/employee/

Type	Params	Values
POST	sortBy	Map<String,String>

#### map

Contains the two keys for sorting employee records. sortBy: attribute\_name, orderBy: order(ASC or DESC)

#### Response

Status	Response
200	<p><b>A List containing the Employees is given</b></p> <p>Example response:-</p> <pre>[   {     "fname": "Ashish",     "lname": "S",     "email": "ashish.s@gmail.com",     "address": "77 Madison",     "phone": "8932898989",     "hireDate": "2018-01-01",     "position": "Software Tester",     "salary": 70000,     "department": "Testing",</pre>

	<pre>         "id": 4,         "photo": null       },       {         "fname": "Rakesh",         "lname": "Mahodya",         "email": "rakesh.maho@gmail.com",         "address": "88 Churchil Street",         "phone": "3372237878",         "hireDate": "2018-01-09",         "position": "Software Engineer",         "salary": 70000,         "department": "Development",         "id": 2,         "photo": null       },       {         "fname": "Abhishiek",         "lname": "Chaudhari",         "email": "achaudhari@albany.edu",         "address": "187 Wolf Rd",         "phone": "7728382819",         "hireDate": "2018-02-14",         "position": "Software Engineer",         "salary": 65000,         "department": "Development",         "id": 3,         "photo": null       },       {         "fname": "Ashish",         "lname": "Balani",         "email": "ashish.balani.94@gmail.com",         "address": "99 Hawthorne Ave",         "phone": "3392939486",         "hireDate": "2018-02-16",         "position": "CEO",         "salary": 2000000,         "department": "Board",         "id": 1,         "photo": null       }     ]   } </pre>
--	---

	]
500	{"error": "Something went wrong. Please try again later."}

## 4. employee

Gets a single employee by his id

### Request

Method	URL
POST	api/employee/emp

Type	Params	Values
POST	map	Map<String,String>

map

Contains the employee id whose information is required

### Response

Status	Response
200	<p><b>A single Employee is given</b></p> <pre>{   "fname": "Ashish",   "lname": "Balani",   "email": "ashish.balani.94@gmail.com",   "address": "99 Hawthorne Ave",   "phone": "3392939486",   "hireDate": "2018-02-16",   "position": "CEO",</pre>



	<pre>    "salary": 2000000,     "department": "Board",     "id": 1,     "photo": null   } ]</pre>
500	<pre>{"error": "Something went wrong. Please try again later."}</pre>

# Glossary

## Conventions

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under 'Responses' for each method. Only one of them is issued per request server.
- All response are in JSON format.
- All request parameters are mandatory unless explicitly marked as **[optional]**
- The type of values accepted for a *request* parameter are shown the the values column like this **[10|<any number>]**. The | symbol means *OR*. If the parameter is **[optional]**, the default value is shown in blue bold text, as **10** is written in **[10|<any number>]**.

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

**2XX** - Success of some kind

**4XX** - Error occurred in client's part

**5XX** - Error occurred in server's part

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict

412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable