Salary Disbursement API

Table of Contents

Overview

This an API to disburse salaries to employees of companies. Each registered company has an Admin who can review payment information of employees that are uploaded by Managers of said company. The entire system is overseen by a Superuser.

The Superuser can register companies and create Admin users for these companies. The Admin users can create Managers. The Managers are responsible for uploading csv files containing employee payment information. The Admin can edit or delete records of this information and then approve of the beneficiary which schedules transactions.

Permission Classes

Class	Condition
AllowAny	User is anyone, even unauthenticated.
IsAuthenticated	User is authenticated.

Class	Condition
IsSuperuser	User is a Superuser.
IsAdmin	User is an Admin.
IsManager	User is a Manager.
IsVerified	User changed initially provided password.
IsSameCompany	User belongs to the same company as the target.
IsSameUser	User is the same as the target.

Base URL

The base URL for all API requests is:
`https://127.0.0.1:8000`

Basic Information

Authorization: Bearer Token
Request and response type: JSON

Endpoints

User

These endpoints are related to the users and facilitate functions such as register, login, profile view etc.

POST /user/register_super

Registers the Superuser who has access to everything within the system.

Permission Classes

AllowAny

Body

- `username` The username of the superuser. **Required**, Type = **Character**
- `email` The email of the superuser. **Required**, Type = **Email**
- `password` The password of the superuser. **Required**, Type = **Character**
- `password2` The confirmation password of the superuser. `password` and `password2` must match. **Required**, Type = **Character**

Response

- `id` The user id of the superuser.
- `username` The username of the superuser
- `email` The email of the superuser

Example

Request:

```
{
    "username": "Super",
    "email": "super@saldis.com",
    "password": "drf@2024!",
    "password2": "drf@2024!"
}
```

Response:

```
{
    "id": 1,
    "username": "Super",
    "email": "super@saldis.com"
}
```

Status Codes

This API uses the following error codes:

- `400 Bad Request`: The request was malformed or missing required properties.
- `201 Created` Successfully registered user in database.

POST /user/register/

Registers a user. Superuser or Admin has permission to access this endpoint. Superuser can register Admins and Admins register Managers.

Permission Classes

IsAuthenticated & (IsSuperuser | (IsVerified & IsSameCompany & IsAdmin))

Body

- `username` The username of the superuser. **Required**, Type = **Character**
- `email` The email of the superuser. **Required**, Type = **Email**
- `password` The password of the superuser. **Required**, Type = **Character**
- `password2` The confirmation password of the superuser. `password` and `password2` must match. **Required**, Type = **Character**
- `company_id` The ID of the company the user belongs to. **Required**, Type = **Integer**
- `group` The group the user belongs to. **Values** =(Admin, Manager) **Required**, Type = Character
- `first_name` The first name of the user. **Required**, Type = Character
- `last_name` The last name of the user. **Required**, Type = Character
- `dob` The date of birth of the user. **Required**, Type = Date
- `gender` The gender of the user. **Required**, Type = Character
- `mobile` The mobile number of the user. **Required**, Type = Character
- `address` The address of the user. **Required**, Type = Character

Response

- `id` The user id of the superuser
- `username` The username of the superuser
- `email` The email of the superuser
- `company_id` The ID of the company the user belongs to

Example

Request:

```
{
    "username": "Mashiat",
    "email": "mashiat@gmail.com",
    "password": "drf@2024!",
    "password2": "drf@2024!",
    "first_name": "Mashiat",
```

```
    "last_name": "Hasin",
    "gender": "Female",
    "mobile": "01575176218",
    "dob": "2001-06-19",
    "address": "Ramna",
    "group": "Admin",
    "company_id": 1
}
```

Response:

```
{
 "id": 2,
    "username": "Mashiat",
    "email": "mashiat@gmail.com",
    "company_id": 1
}
```

Status Codes

This API uses the following error codes:

- `400 Bad Request`: The request was malformed or missing required properties.
- `201 Created`: Successfully registered user in database.

POST /user/login/

Provides a JWT access token to the user. Access tokens expire in 1 hour.

Permission Classes

AllowAny

Body

- `username` The username of the user. **Required**, Type = **Character**
- `password` The password of the user. **Required**, Type = **Character**

Response

- `redirect_url` A redirect URL to change default password. Only shown to users who haven't changed their default passwords. **Optional**
- `access` The access token.
- `refresh` The refresh token.

Example

Request:

```
{
    "username": "Mashiat",
    "password": "drf@2024!"
}
```

Response:

```
{
    "redirect_url": "/user/reset_password/",
    "access": "eyJhbG...iAwPY",
```

```
    "refresh": "eyJhbG...eyIw8"
}
```

```
{
    "detail": "No active account found with the given credentials"
}
```


Status Codes

This API uses the following error codes:

- `401 Unauthorized`: The provided credentials are invalid.
- `200 OK`: Retrieved token successfully.

POST /user/token_refresh/

Provides a new access token to the user using their refresh token. Refresh Tokens expire in 1 day.

Permission Classes

AllowAny

Body

- `refresh` The refresh token of the superuser. **Required**, Type = **Character**

Response

- `access` The access token of the user.
- `detail` Message if action was unsuccessful. **Optional**
- `code` Error code. **Optional**

Example

Request:


```
{
    "refresh": "eyJhbG...DU9lw"
}
```


Response:


```
{
    "access": "eyJhbG...ScNpQ"
}
```

```
{
        "detail": "Token is invalid or expired",
        "code": "token_not_valid"
}
```


Status Codes

This API uses the following error codes:

- `401 Unauthorized`: The provided refresh token has expired.
- `200 OK`: Retrieved token successfully.

PATCH /user/reset_password/

Resets the user's password. The new password cannot be the same as the old password.

Permission Classes

IsAuthenticated

Body

- `username` The username of the user. **Required**, Type = **Character**
- `password` The new password of the user. **Required**, Type = **Character**
- `password2` The new confirmation password of the user. **Required**, Type = **Character**

Response

- `detail` Confirmation of successful password reset.

Example

Request:

```
{
        "username": "Mashiat",
        "password": "drf@2024!",
        "password2": "drf@2024!"
}
```

Response:

```
{
        "detail": "Password updated successfully"
}
```

Status Codes

This API uses the following error codes:

- `400 Bad Request`: The request was malformed or missing required properties.
- `200 OK`: Password changed successfully.

GET /user/profile/<int:id>/

Retrieves a user's profile.

Permission Classes

IsAuthenticated & (IsSameCompany | IsSuperuser) & IsVerified

URL Converter Parameters

- `id` The id of the user to retrieve profile from. **Required**, Type = **Integer**

Response

- `id` The ID of the user.
- `username` The username of the user.
- `email` The email of the user.
- `company_id` The ID of the company the user belongs to.

- `group` The group the user belongs to.
- `first_name` The first name of the user.
- `last_name` The last name of the user.
- `dob` The date of birth of the user.
- `gender` The gender of the user.
- `mobile` The mobile number of the user.
- `address` The address of the user.

Example

URL:

```
http://127.0.0.1:8000/user/profile/2/
```

Response:


```
{
  "id": 2,
  "username": "Noshin",
  "email": "noshin@gmail.com",
  "company_id": 1,
  "group": [
      "Manager"
  ],
  "first_name": "Noshin",
  "last_name": "Tarannum",
  "gender": "Female",
  "dob": "2000-03-13",
  "mobile": "01748372946",
  "address": "Ramna"
}
```


Status Codes

This API uses the following error codes:

- `404 Not Found`: The requested user does not exist.
- `200 OK`: Retrieved user successfully.

GET /user/profile/all/

Retrieves all users. Only the superuser has access to this endpoint.

Permission Classes

IsAuthenticated & IsSuperuser

Parameters

- `page_no` The page number to retrieve. **Optional**, Default = 1
- `page_size` The number of records per page. **Optional**, Default = 10

Response

Same as `/user/profile/<int:id>` **Multiple**

Example

URL:

```
http://127.0.0.1:8000/user/profile/all/?page_no=1&page_size=3
```

Response:


```
[
  {
      "id": 1,
      "username": "super",
      "email": "super@saldis.come",
      "company_id": 0
  },
  {
      "id": 3,
      "username": "Mashiat",
      "email": "mashiat@gmail.com",
      "company_id": 1,
      "group": [
          "Admin"
      ],
      "first_name": "Mashiat",
      "last_name": "Hasin",
      "gender": "Female",
      "dob": "2001-06-19",
      "mobile": "01575176218",
      "address": "Eskaton Garden Rd"
  },
  {
      "id": 2,
      "username": "Noshin",
      "email": "noshin@gmail.com",
      "company_id": 1,
      "group": [
          "Manager"
      ],
      "first_name": "Noshin",
      "last_name": "Tarannum",
      "gender": "Female",
      "dob": "2000-03-13",
      "mobile": "01284287319",
      "address": "Ramna"
  }
]
```


Status Codes

This API uses the following error codes:

- `404 Not Found`: The requested user does not exist.
- `200 OK`: Retrieved user successfully.

PATCH /user/profile/<int:id>/update/

Updates a user's information. General information that are not vital to the system (id, company_id) can be updated. Password cannot be updated from this endpoint. Please refer to the `user/reset_password` endpoint.

Permission Classes

IsAuthenticated & (IsSuperuser | (IsVerified & ((IsAdmin & IsSameCompany) | IsSameUser)))

URL Converter Parameters

- `id` The id of the user to retrieve profile from. **Required**, Type = **Integer**

Body

same as `/user/profile/<int:id>` endpoint. However, every field is **optional**

Response

same as `/user/profile/<int:id>` endpoint.

Example

URL:

```
http://127.0.0.1:8000/user/profile/2/update/
```

Request:

```
{
        "username": "Nosh",
        "mobile": "017629472981"
}
```

Response:

```
{
  "id": 2,
  "username": "Nosh",
  "email": "noshin@gmail.com",
  "company_id": 1,
  "group": [
      "Manager"
  ],
  "first_name": "Noshin",
  "last_name": "Tarannum",
  "gender": "Female",
  "dob": "2000-03-13",
  "mobile": "01762947298",
  "address": "Ramna"
}
```

Status Codes

This API uses the following error codes:

- `404 Not Found`: The user does not exist.
- `400 Bad Request`: The properties were malformed or not provided.
- `200 OK`: Retrieved user successfully.

Company

These endpoints are related to the companies and facilitate functions such as registering companies, viewing or editing companies etc.

POST /company/register/

Registers a company. Only the Superuser can register companies.

Permission Classes

IsSuperuser

Properties

- `organization` The name of the organization. **Required**, Type = **Character**
- `founded_year` The year the organization was founded in. **Required**, Type = **Integer**
- `description` A short description of the organization. **Required**, Type = **Text**
- `is_active` Whether the organization is active or not. **Required**, Type = **Boolean**
- `industry` The domain of the organization. **Required**, Type = **Character**
- `website` Website link of the organization. **Required**, Type = **URL**
- `email` Email address of the organization. **Required**, Type = **Email**

Response

- `id` The ID of the registered organization.
- `organization` The name of the organization.
- `founded_year` The year the organization was founded in.
- `description` A short description of the organization.
- `is_active` Whether the organization is active or not.
- `industry` The domain of the organization.
- `website` Website link of the organization.
- `email` Email address of the organization.

Example

Request:

```
{
        "organization": "Square",
        "founded_year": 1990,
        "description": "None",
        "is_active": true,
        "industry": "pharmaseuticals",
        "website": "https://www.square.com/",
        "email": "contact@square.come"
}
```

Response:

```
{
        "id": 1,
        "organization": "Square",
        "founded_year": 1990,
        "description": "None",
        "is_active": true,
        "industry": "pharmaseuticals",
        "website": "https://www.square.com/",
        "email": "contact@square.come"
}
```

Status Codes

This API uses the following error codes:

- `400 Bad Request`: The request was malformed or missing required properties.
- `200 OK`: Registered company successfully.
- `500 Internal Server Error`: Integrity error. Company with same name exists.

GET /company/<int:id>/

Retrieves the information of a specific company.

Permission Classes

IsAuthenticated & ((IsVerified & IsSameCompany) | IsSuperuser)

Parameters

- `page_no` The page number to retrieve. **Optional**, Default = 1
- `page_size` The number of records per page. **Optional**, Default = 10

Response

- `id` The ID of the registered organization.
- `organization` The name of the organization.
- `founded_year` The year the organization was founded in.
- `description` A short description of the organization.
- `is_active` Whether the organization is active or not.
- `industry` The domain of the organization.
- `website` Website link of the organization.
- `email` Email address of the organization.
- `admin` A list of admin users of this company.
- `manager` A list of manager users of this company.

Example

URL:

```
http://127.0.0.1:8000/company/1/
```

Response:

```
{
  "id": 1,
  "organization": "Square",
  "founded_year": 1976,
  "description": "None",
  "is_active": true,
  "industry": "pharmaseuticals",
  "website": "http://www.square.com",
  "email": "contact@square.com",
  "admin": [
      3
  ],
  "manager": [
      2
  ]
}
```

Status Codes

This API uses the following error codes:

- `200 Ok`: The response was returned successfully.
- `404 Not Found`: The requested company does not exist.

GET /company/all/

Retrieves a list of all the registered companies. Only the Superuser can access this endpoint.

Parameters

- `page_no` The page number to retrieve. **Optional**, Default = 1
- `page_size` The number of records per page. **Optional**, Default = 10

Response

same as `/company/<int:id>/` endpoint. **Multiple**

Example

Response:


```
[
  {
      "id": 1,
      "organization": "Square",
      "founded_year": 1976,
      "description": "None",
      "is_active": true,
      "industry": "pharmaseuticals",
      "website": "http://www.square.com",
      "email": "contact@square.com",
      "admin": [
          3
      ],
      "manager": [
          2
      ]
  },
  {
      "id": 2,
      "organization": "Aarong",
      "founded_year": 1976,
      "description": "None",
      "is_active": true,
      "industry": "Clothing",
      "website": "http://www.aarong.com",
      "email": "contact@aarong.com",
      "admin": [
          4
      ],
      "manager": [
          5
      ]
  }
]
```


Status Codes

This API uses the following error codes:

- `200 Ok`: The response was returned successfully.
- `204 No Content`: There are no registered companies.

- `404 Not Found`: The requested page does not exist.

PATCH /company/<int:id>/update/

Updates the information of a company. Only the Superuser and company Admin can access this endpoint. General information that are not vital to the system (id) can be updated.

Permission Classes

IsAuthenticated & (IsAdmin & IsVerified & IsSameCompany) | IsSuperuser

Body

same as `/comapny/<int:id>/` endpoint. However, every field is **optional**

Response

same as `/company/<int:id>/` endpoint.

Example

URL:

```
http://127.0.0.1:8000/company/1/update/
```

Request:

```
{
        "organization": "Square Ltd.",
        "description": "This is a renowned company in Bangladesh."
}
```

Response:

```
{
  "id": 1,
  "organization": "Square Ltd.",
  "founded_year": 1976,
  "description": "This is a renowned company in Bangladesh.",
  "is_active": true,
  "industry": "pharmaseuticals",
  "website": "http://www.square.com",
  "email": "contact@square.com"
}
```

Status Codes

This API uses the following error codes:

- `200 Ok`: The response was returned successfully.
- `400 No Bad Request`: The request body is malformed.
- `404 Not Found`: The requested page does not exist.

Salary

These endpoints are related to the salary and payment information. It deals with functions such as uploading files with payment information, editing salary/payment information, approving for transaction etc.

GET /salary/file/upload/

Upload a new file containing employee payment information. Immediately after the file is uploaded, the information gets stored in a salary table **asynchronously** using **celery**. This ensures that the system remains in service during the storage. No new fields are created if past employee information exists. Old payment information of the employees just get updated in that case.

Permission Classes

IsAuthenticated & IsManager & IsVerified

Body

- `file` The file to be uploaded. **Required**, Type = **File**

Response

- `id` The ID of the uploaded file.
- `filepath` The path of the uploaded file.
- `company_id` The ID of the organization the file belongs to.
- `uploader_id` The ID of the user who uploaded the file.
- `created` The date of creation.
- `is_approved` Whether the file was approved by the admin or not.
- `is_complete` Whether transaction from this file were made.

Example

Response:


```
{
  "id": 3,
  "filepath": "/media/payment_info.csv",
  "company_id": 1,
  "uploader_id": 2,
  "created": "2024-03-14T14:03:49.100838+06:00",
  "is_approved": false,
  "is_complete": false
}
```


Status Codes

This API uses the following error codes:

- `400 Bad Request` : The request was malformed or missing required properties.
- `201 Created` : Uploaded file successfully.

GET /salary/beneficiary/all/

View all the beneficiaries of own company.

Permission Classes

IsAuthenticated & IsSameCompany & IsVerified

Parameters

- `page_no` The page number to retrieve. **Optional**, Default = 1
- `page_size` The number of records per page. **Optional**, Default = 10

Response

Same as `/salary/file/upload/` **Multiple**

Example

Response:


```
[
  {
      "id": 1,
      "filepath": "/media/payment_info.csv",
      "company_id": 1,
      "uploader_id": 2,
      "created": "2024-03-12T14:24:18.519543+06:00",
      "is_approved": true,
      "is_complete": true
  },
  {
      "id": 2,
      "filepath": "/media/payment_info.csv",
      "company_id": 1,
      "uploader_id": 2,
      "created": "2024-03-14T14:01:51.496526+06:00",
      "is_approved": false,
      "is_complete": false
  },
  {
      "id": 3,
      "filepath": "/media/payment_info.csv",
      "company_id": 1,
      "uploader_id": 2,
      "created": "2024-03-14T14:03:49.100838+06:00",
      "is_approved": false,
      "is_complete": false
  }
]
```


Status Codes

This API uses the following error codes:

- `200 Ok`: The response was returned successfully.
- `204 No Content`: The company has no beneficiaries.
- `404 Not Found`: The requested page does not exist.

GET /salary/beneficiary/<int:id/

View information of a specific beneficiary from the same company.

Permission Classes

IsAuthenticated & IsSameCompany & IsVerified

URL converter parameter

- `id` The ID of the beneficiary to be retrieved.

Response

- same as `/salary/file/upload/`
- `schedule_time` The date and time when the salary mentioned in the beneficiary will be disbursed.

Example

URL:

```
http://127.0.0.1:8000/salary/beneficiary/1/
```

Response:


```
{
    "id": 1,
    "filepath": "/media/payment_info.csv",
    "company_id": 1,
    "uploader_id": 2,
    "created": "2024-03-12T14:24:18.519543+06:00",
    "is_approved": true,
    "is_complete": true
}
```


Status Codes

This API uses the following error codes:

- `404 Not Found` : The request beneficiary does not exist.
- `200 OK` : Retrieved beneficiary successfully.

PATCH /salary/beneficiary/<int:id/approve/

Approves a beneficiary. This action can be performed by an Admin only. The Admin can approve of the beneficiary after making any changes and needs to provide a `schedule_time` As soon as the beneficiary is approved, the disbursement is scheduled to occur at the specified time. The transaction happens **asynchronously** in the background to keep the system in service.

Permission Classes

IsAuthenticated & IsSameCompany & IsVerified & IsAdmin

URL converter parameter

- `id` The ID of the beneficiary to be approved.

Request

- `is_approved` Whether the beneficiary is approved. **Required**, Type = **Boolean**
- `schedule_time` The The date and time when the salary mentioned in the beneficiary will be disbursed. **Required**, Type = **Datetime**

Response

- same as `/salary/file/upload/`
- `schedule_time` The date and time when the salary mentioned in the beneficiary will be disbursed.

Example

URL:

```
http://127.0.0.1:8000/salary/beneficiary/1/approve/
```

Request:

```
{
        "is_approved": true,
        "schedule_time": "2024-03-22T08:24:18.519543Z"
}
```


Response:


```
{
    "id": 1,
    "filepath": "/media/payment_info.csv",
    "company_id": 1,
    "uploader_id": 2,
    "created": "2024-03-12T14:24:18.519543+06:00",
    "is_approved": true,
    "is_complete": true,
    "schedule_time": "2024-03-22T08:24:18.519543+06:00"
}
```


Status Codes

This API uses the following error codes:

- `400 Bad Request`: The request was malformed or missing required properties.
- `200 OK`: Successfully approved the beneficiary successfully.

GET /salary/all/

View all the payment information of employees of own company.

Permission Classes

IsAuthenticated & IsSameCompany & IsVerified

Parameters

- `page_no` The page number to retrieve. **Optional**, Default = 1
- `page_size` The number of records per page. **Optional**, Default = 10

Response

- `id` The ID of the payment information.
- `employee_id` The ID of the employee.
- `wallet_no` The wallet number of the employee.
- `amount` The amount to be paid to the employee.

Example

Response:


```
[
  {
      "id": 43,
      "employee_id": "44",
      "wallet_no": 13799,
      "amount": 136774,
      "beneficiary_id": 2
  },
  {
      "id": 45,
```

```
        "employee_id": "43",
        "wallet_no": 13919,
        "amount": 141029,
        "beneficiary_id": 2
    },
    {
        "id": 44,
        "employee_id": "46",
        "wallet_no": 10909,
        "amount": 73006,
        "beneficiary_id": 2
    },
    {
        "id": 46,
        "employee_id": "45",
        "wallet_no": 17636,
        "amount": 104881,
        "beneficiary_id": 2
    },
    {
        "id": 48,
        "employee_id": "47",
        "wallet_no": 19658,
        "amount": 152096,
        "beneficiary_id": 2
    }
]
```


Status Codes

This API uses the following error codes:

- `200 Ok`: The response was returned successfully.
- `204 No Content`: There are no registered companies.
- `404 Not Found`: The requested page does not exist.

GET /salary/<int:id/

View details of a specific salary record.

Permission Classes

IsAuthenticated & IsSameCompany & IsVerified & IsAdmin

URL converter parameter

- `id` The ID of the beneficiary to be retrieved.

Response

- same as `/salary/file/upload/` **Multiple = False**

Status Codes

This API uses the following error codes:

- `404 Not Found`: The requested salary information does not exist.
- `200 OK`: Retrieved salary information successfully.

PATCH /salary/<int:id/

Edit details of a specific salary record.

Permission Classes

IsAuthenticated & IsSameCompany & IsVerified & IsAdmin

URL converter parameter

- `id` The ID of the beneficiary to be retrieved.

Body

- `employee_id` The ID of the employee. **Optional**, Type= **Integer**
- `wallet_no` The wallet number of the employee. **Optional**, Type= **Integer**
- `amount` The amount to be paid to the employee. **Optional**, Type= **Integer**

Response

- same as `/salary/file/upload/`

Example

URL:

```
http://127.0.0.1:8000/salary/43/
```

Request:

```
{
        "amount": 1000
}
```

Response:

```
{
        "id": 43,
        "employee_id": "44",
        "wallet_no": 13799,
        "amount": 1000,
        "beneficiary_id": 2
    }
```

Status Codes

This API uses the following error codes:

- `404 Not Found`: The requested salary information does not exist.
- `400 Bad Request`: The request was malformed or missing required properties.
- `200 OK`: Edited salary information successfully.

DEL /salary/<int:id/

Deletes a specific salary record.

Permission Classes

IsAuthenticated & IsSameCompany & IsVerified & IsAdmin

URL converter parameter

- `id` The ID of the beneficiary to be retrieved.

Response

- `detail` Informs whether deletion was successful.

Status Codes

This API uses the following error codes:

- `400 Bad Request` : The record does not exist and cannot be deleted.
- `200 OK` : Deleted record successfully.