

API Documentation

1. Introduction

This Project API is a web service which allows you to use our Machine Learning platform from your website, software or mobile application. The API gives access to all of the supported functions of our service.

2. Authentication

Token-based Authentication: Our API uses Laravel Sanctum for user authentication. You must obtain a valid token to access our API's data. Here's an overview of the authentication process:

3. Obtaining an API Token

- ✓ **Step 1 : Log In:** users need to log in using their credentials.
- ✓ **Step 2 : Obtain a Token:** Upon successful login, users will receive a token that they must include in their API requests' headers. The token serves as proof of authentication.

HTTP Method: POST

URL: <site_url>/api/login

Parameter Name	Description
email	Required – (email)
password	required

Response JSON:

```
{
  "status": 200,
  "token": "1|b1U4ei568hhXnvPcfMCdT4NyA3HCLa5X9onCss0B9f5a9642",
  "message": "Api Token Created Successfully"
}
```

4. API Request Headers

To access our API data, you must include the following header in your HTTP requests:

Key	Value
Authorization	Bearer <API_TOKEN>
Accept	application/json

Replace API_TOKEN with the token you received during the login process.

Company - CRUD

1. Show All Companies: -

HTTP Method: POST

URL: <site_url>/api/companies

2. Show One Company: -

HTTP Method: POST

URL: <site_url>/api/company

Parameter Name	Description
company_id	Required – (number)

Response JSON:

```

{
  "status": 200,
  "data": {
    "company_id": 2,
    "name": "Sanoria",
    "email": "bvvv@gf.com",
    "logo": "1699462558.jpg",
    "website": "jhhuh",
    "created_at": "2023-11-08T16:55:58.000000Z",
    "updated_at": "2023-11-09T03:57:20.000000Z",
    "logoUrl": "http://127.0.0.1:8000/storage/logos/1699462558.jpg"
  }
}

```

3. Create New Company:-

HTTP Method: POST

URL: <site_url>/api/add-company

Parameter Name	Description
name	Required
email	Required – (email)
logo	Required – (image)
website	Required

Response JSON:

```

{
  "status": 200,
  "message": "Company Added Successfully"
}

```

4. Edit Company: -

HTTP Method: POST

URL: <site_url>/api/edit-company

Parameter Name	Description
company_id	Required – (number)
name	Required
email	Required – (email)
logo	(image)
website	Required

Response JSON:

```
{
  "status": 200,
  "message": "Company Updated Successfully"
}
```

5. Delete Company: -

HTTP Method: POST

URL: <site_url>/api/delete-company

Parameter Name	Description
company_id	Required – (number)

Response JSON:

```
{
  "status": 200,
  "message": "Company and worked Employees are Deleted!"
}
```

Employee - CRUD

1. Show All Employees: -

HTTP Method: POST

URL: <site_url>/api/employees

2. Show One Employee: -

HTTP Method: POST

URL: <site_url>/api/employee

Parameter Name	Description
employee_id	Required – (number)

Response JSON:

```
{
  "status": 200,
  "data": {
    "employee_id": 2,
    "fname": "sumesh",
    "lname": "Karthik",
    "company_id": 4,
    "email": "fgdfg@m.com",
    "phone": 152454,
    "created_at": "2023-11-09T08:39:30.000000Z",
    "updated_at": "2023-11-09T08:39:30.000000Z"
  }
}
```

3. Create New Employee: -

HTTP Method: POST

URL: <site_url>/api/add-employee

Parameter Name	Description
fname	Required
lname	Required
email	Required – (email)
phone	Required – (number)
company_id	Required – (number)

Response JSON:

```
{  
  "status": 200,  
  "message": "Employee Added Successfully"  
}
```

4. Edit Employee: -

HTTP Method: POST

URL: <site_url>/api/edit-employee

Parameter Name	Description
company_id	Required – (number)
employee_id	Required – (number)
name	Required
email	Required – (email)
logo	Required – (image)
website	Required

Response JSON:

```
{
  "status": 200,
  "message": "Employee Updated Successfully"
}
```

5. Delete Employee: –

HTTP Method: POST

URL: <site_url>/api/delete-employee

Parameter Name	Description
employee_id	Required – (number)

Response JSON:

```
{
  "status": 200,
  "message": "Employee Deleted Successfully!"
}
```

Logout User:

HTTP Method: POST

URL: <site_url>/api/logout

Response JSON:

```
{
  "status": 200,
  "message": "User Logout Successfully"
}
```

Conclusion:

Our API relies on Laravel Sanctum for secure token-based authentication. By following the steps outlined in this documentation, you can access our data securely.