

Enterkey leave Application API description

SI	API	Method	Action	Description
1	/register	POST	Register a new user	Creates a new user record in the system.
2	/login	POST	Authenticate and generate an access token.	Establishes a secure user session, providing access to protected resources.
3	/dashboard	POST	Retrieve authenticated user details.	Provides information about the currently authenticated user.
4	/auth/change-password	POST	Change the password for the authenticated user	Updates the password associated with the user account.
5	/logout	POST	Log out the authenticated user (invalidate API token)	Terminates the user's session, requiring re authentication for access.
6	/users	GET	Retrieve a paginated list of users, including company details and roles.	Returns user data associated with their roles and company affiliations.
7	/upload/image	POST	Uploads and updates the user's profile picture.	Associates the uploaded image as the user's profile picture.
8	/upload/sign	POST	Uploads and updates the user's signature.	Associates the uploaded image as the user's signature.
9	/employees	GET	Retrieve a paginated list of employees with details like designation, branch, and department.	Displays information about employees, considering roles and company affiliations.
10	/employees/create	POST	Create a new employee and associate them with a user.	Establishes a relationship between the user and employee records.
11	/employees/update/{id}	POST	Update details of an existing employee based on their ID.	Modifies the information associated with a specific employee.

12	/employees/{id}	POST	Delete an employee and their associated user.	Removes the records of a specific employee and their user account.
13	/employee/reset/password	POST	Reset the password for a specific user.	Updates the password associated with the user account.
14	/employee/profile	GET	Retrieve the profile details of the currently authenticated employee.	Returns details like designation, branch, and department for the authenticated employee.
15	/employee/update/profile	POST	Update the profile details of the currently authenticated employee.	Modifies the personal information and signature of the authenticated employee.
16	/employee/total-leave/{id}	GET	Retrieve the total approved leave days for a specific employee.	Provides information about the approved leave days for a particular employee.
17	/employee/total-casual-leave/{id}	GET	Retrieve the total approved casual leave days and remaining days for a specific employee.	Displays information about casual leave days for a particular employee.
18	/employee/details/{id}	GET	Retrieve detailed information about a specific employee based on their ID.	Displays comprehensive details, including designation, branch, department, and user information.
19	/company/create	POST	Create a new company with details such as name, contact address, contact number, contact person, contact email, BIN, and TIN.	Establishes a new company record with the associated details.
20	/company/update/{id}	POST	Update details of an existing company based on its ID.	Modifies the information associated with a specific company.
21	/company/delete/{id}	POST	Delete a company based on its ID.	Removes the records of a specific company and its associated branches, given certain conditions.

22	/company/{id}	GET	Retrieve details of a specific company based on its ID.	Displays information about a specific company.
23	/companies	GET	Retrieve a paginated list of companies.	Displays a list of companies, considering user roles and company affiliations.
24	/branch/create	POST	Create a new branch with details such as branch name	Establishes a new branch associated with the logged-in user's company.
25	/branch/update/{id}	POST	Update details of an existing branch based on its ID.	Modifies the information associated with a specific branch
26	/branch/delete/{id}	POST	Delete a branch based on its ID.	Removes the records of a specific branch, given certain conditions (e.g., no associated employees).
27	/branches	GET	Retrieve a paginated list of branches.	Displays a list of branches associated with the logged-in user's company, considering user roles and company affiliations
28	/department/create	POST	Create a new department with details such as department name	Establishes a new department and records the action in the activity log.
29	/department/update/{id}	POST	Update details of an existing department based on its ID	Modifies the information associated with a specific department
30	/department/delete/{id}	POST	Delete a department based on its ID	Removes the records of a specific department, given certain conditions (e.g., no associated employees).
31	/departments	GET	Retrieve a paginated list of departments.	Displays a list of departments, allowing pagination to manage large datasets.
32	/grade/create	POST	Create a new grade with details such as grade name.	Establishes a new grade and records the action in the activity log

33	/grade/update/{id}:	POST	Update details of an existing grade based on its ID	Modifies the information associated with a specific grade
34	/grade/delete/{id}	POST	Delete a grade based on its ID	Removes the records of a specific grade, given certain conditions (e.g., no associated designations).
35	/grades	POST	Retrieve a paginated list of grades.	Displays a list of grades, allowing pagination to manage large datasets.
36	/designation/create	POST	Create a new designation with details such as designation name and associated grade	Establishes a new designation and records the action in the activity log.
37	/designation/update/{id}	POST	Update details of an existing designation based on its ID, including the designation name and associated grade.	Modifies the information associated with a specific designation
38	/designation/delete/{id}	POST	Delete a designation based on its ID	Removes the records of a specific designation, given certain conditions (e.g., no associated employees).
39	/designations	GET	Retrieve a paginated list of designations with associated grade details.	Displays a list of designations, including details about the associated grade, allowing pagination to manage large datasets.
40	/leave/create:	POST	Create a new application	Submit a request for a leave application. The system validates the provided details, including leave type, start and end dates, and stay location. If the application meets the

				criteria, it is stored in the database. The system also checks for the maximum allowable leave duration and provides appropriate messages. Notifications are sent to the designated reviewer and the applicant upon successful submission.
41	/leave/select-approver/{id}	POST	Set Approver for Leave Application	This API lets a logged-in user choose an approver for a specific leave request (identified by {id}). It checks if the user is logged in; if not, it says 'Login first' (401 Unauthorised). It then finds the employee details and the specified leave request. If not found, it says 'Application not found' (404 Not Found). If found, it updates the approver and logs the action. Success gets a 'Approver is selected' (200 OK) message with the updated leave request data.
42	/leave/employee-list	GET	Retrieve Approvers List	This API fetches a list of leave approvers in the same company as the logged-in user. It verifies the user is logged in (401 Unauthorized if not). Using a left join on 'employee,' 'users,' and 'designation' tables, it compiles details about each approver, excluding the logged-in user. The result includes user, employee, and designation information for each leave approver.
43	/leave/recorder-list	GET	Retrieve Recorders	This API fetches a list of

			List	leave recorders in the same company as the logged-in user. It checks if the user is logged in (401 Unauthorized if not). Using a left join on 'employee,' 'users,' and 'designation' tables, it compiles details about each recorder, excluding the logged-in user. The result includes user, employee, and designation information for each leave recorder.
44	/leave/application-list	GET	Retrieve Leave Applications List	This API fetches a list of leave applications designated to the logged-in user as the approver. It checks if the user is logged in (401 Unauthorized if not). It retrieves the user's employee ID and fetches leave applications where 'approval_id' matches the employee ID. The result includes details like application ID, leave type, start and end dates for each assigned leave application.
45	/leave/own-application-list	GET	Retrieve Own Leave Applications List	This API fetches a list of leave applications submitted by the logged-in user. It verifies the user is logged in (401 Unauthorized if not). By retrieving the user's employee ID and fetching leave applications where 'employee_id' matches, it ensures only the user's submissions are included. The result provides details like application ID, leave type, start and end dates for each submitted leave

				application.
46	/leave/applied-list	POST	Retrieve Applied Leave Applications List	<p>This API gets a list of leave applications based on specified criteria. It first checks if the user is logged in (401 Unauthorized if not). After getting user details, it constructs a query using left joins with related tables for sender, approver, reviewer, leave type, and status. The query includes conditions like leave type, selected employee, status, start date, and end date. The response, displayed in pages, includes details for each leave application, such as user, company, and other relevant info.</p>
47	/leave/application-approved/{id}	POST	Approve Leave Application	<p>This API allows authenticated approvers to approve a leave application by {id}. It checks the user's login status, retrieves approver details, and locates the specified application matching the approver's ID. If not found, a 404 Not Found response is returned. The API updates the application's approved start date, end date, and total days based on parameters, defaulting to the reviewer's or applicant's details as needed. Comments from the request are stored, setting the application status to 2 (approved). A message is sent to the applicant, and "ApplicationPassingHistory" table is updated. Laravel activity logging</p>

				records the approval action.
48	/leave/application-reject/{id}	POST	Reject Application Leave	<p>This API allows authenticated users (approvers or reviewers) to reject a leave application identified by {id}. It verifies user login status (returns 401 Unauthorized if not logged in) and retrieves associated employee details. The API then locates the specified leave application; if not found, it returns a 404 Not Found response. The application status is updated to 4 (rejected), saving the changes. Additionally, a new entry in the ApplicationPassingHistory table is created with the rejection status and comments from the request. A rejection message is sent to the applicant, and the rejection action is logged using Laravel activity logging.</p>
49	/leave/application-pass/{id}:	POST	Forward Application Leave	<p>This API enables logged-in recorders to forward a leave request (identified by {id}). It checks user login status, retrieves details, and attempts to find the requested application. If the user or application isn't found, it responds with a message accordingly. The API then determines whether the user is an approver or recorder, advancing the application to the next step (approved for approvers, recorded for recorders). It updates</p>

				details and saves changes. A message is sent, and the action is logged using Laravel activity tracking.
50	/leave/application-return/{id}	POST	Return Application Leave	This API allows logged-in users (approvers or recorders) to send a specific leave application (identified by {id}) back to the applicant. It checks user login status, retrieves details, and attempts to find the specified application. If the user or application is not found, it responds with a 404 Not Found. The API updates the application status to 3 (returned) and saves changes, creating a record in 'ApplicationPassingHistory' with provided comments. A return message is sent, and the action is logged using Laravel activity tracking.
51	/leave/application/for-recorder	POST	Get Applications Recorder Leave for	This API retrieves leave applications tailored for users with the recorder type. To access this information, the user must be logged in and possess the recorder type; otherwise, a 401 Unauthorized response with a 'Login first' message is returned. The API fetches the employee details linked to the authenticated user and retrieves pending leave applications designated for recorder review. It includes essential details like sender info, approver details, reviewer information, leave type,

				and status. Additionally, the API offers optional filters to refine the application list based on leave type, selected employee, leave status, start date, and end date.
52	leave/application/for-approver:	POST	Get Applications for Approver	This API caters to users with the approver type, fetching leave applications specifically for their approval. To access this information, users must be logged in with the approver type; otherwise, a 401 Unauthorized response with a 'Login first' message is issued. The API retrieves employee details linked to the authenticated user and fetches pending leave applications awaiting approval. Essential details such as sender info, recorder details, reviewer information, leave type, and status are included. The API also offers optional filters for refining the application list based on leave type, selected employee, leave status, start date, and end date.
53	/leave/details/{id}:	GET	Get Leave Application Details	This API endpoint retrieves comprehensive details for a specific leave application identified by its unique ID. It verifies the existence of the application, responding with a 404 Not Found and an 'Application not found' message if not present. The API gathers information including details about the approver, sender,

				reviewer, leave type, leave status, and additional application-related specifics. Additionally, it calculates the total approved days for the specified employee, considering the current year.
54	/leave/edit-application/{id}	POST	Edit Leave Application Details	This API enables users to edit specific details of their leave application identified by a unique ID. The user must own the application for edits, and the system ensures this by checking the application's existence and association with the authenticated user. If not found, a 404 Not Found response with 'Application not found' is returned. The API accepts a JSON payload with optional parameters for fields like approval_id, reviewer_id, start, end, reason, stay_location, leave_type_id, and comments. It processes start and end dates to calculate total applied days, and checks the total approved days for the current year, imposing constraints based on leave type. If the maximum allowable leave duration is exceeded, an appropriate response is returned.
55	/leave/application-history/{id}	GET	Get Application History	This API endpoint retrieves the application history for a specific leave application identified by its ID. It uses the Application and

				ApplicationPassingHistory models to fetch the relevant data. The history includes details about the sender, receiver, and changes in the application's status.
56	/leave/approve-count:	GET	Approve Count Leave	This API endpoint retrieves the total number of approved leave days for the authenticated user. It calculates and returns the sum of approved_total_days for leave applications associated with the user's employee ID and having a status of 2 (approved).
57	/leave/approve-leave-count	GET	Approve Count Leave with Remaining Days	This API endpoint retrieves the total number of approved leave days for different leave types along with the remaining days. It calculates the sum of approved_total_days for leave applications associated with the user's employee ID, having a status of 2 (approved), and grouped by leave type. It also calculates the remaining days based on the leave type's maximum duration
58	/leave/application-to-me	POST	Get Applications Addressed to the Authenticated User	This API shows a paginated list of leave applications sent to the logged user. It uses models like Application and ApplicationPassingHistory to gather data. The list has details about the approver, sender, reviewer, leave type, and

				status. You can filter the list by leave type, selected employee, leave status, start date, and end date.
59	/activity-log:	GET	Get All Activity Logs	This API endpoint allows authenticated users to access all activity logs. Only super admins have permission to use this endpoint. The logs contain information about various system activities.
60	/leave/advance-report	POST	Get All Application of applicants	The API endpoint generates an advance leave report based on specified parameters. This action accepts a request and requires authentication for access. The report can be filtered by leave type, employee ID, leave status, start and end dates, application date, and company (for users with role_id greater than 1). The response provides a paginated list of applications, including detailed information about senders, approvers, reviewers, leave types, and statuses. Admin can only view applications within their company.
61	/leave/messages	GET	The action involves retrieving messages associated with the authenticated user, including detailed information about senders, receivers, and related applications.	This API gets messages for the logged-in user. You need to be logged in for this action, and the response shows a paginated list of messages with sender, receiver, and application details. Each message includes info about the sender's branch,

				company, designation, and department. It also provides details about the linked application, such as sender, approver, reviewer, leave type, and status.
--	--	--	--	--