

## Simple Mail Transfer Protocol (SMTP)

Database Name :- smtp\_setups

Data Store :- \_id, corporate\_id , smtp\_name, host\_address, username, from\_email\_address , password, method, port , status, created\_at, updated\_at, \_\_v

Frontend Application :- Click on Company name in right corner of the website , then click on company settings → Master Settings → SMTP → Fill the data → Click “Save”

After that

Api Call:-

<https://api.vauras.cloud/api/company/update-smtp-access>

<https://api.vauras.cloud/api/company/get-smtp-list>

Then Mail use in Frontend in :- “Salary” in left side bar → Pay Slip → Fill the data → Click on “SEND VIA @MAIL”

Api Call :-

<https://api.vauras.cloud/api/company/send-payslip-bulk>

Backend Application :-

Database Model:- Model → Admin → SmtSetup.js

Routes:-

1. Routes → Company.js
2. Routes → admin.js

Api Written :- Controller → Admin → SmtsetupController.js

1. get\_smtp\_list
2. add\_smtp\_access
3. update\_smtp\_setup\_data
4. update\_smtp\_setup\_status
5. delete\_smtp\_setup
6. send\_email (function)

Explanation :-

get\_smtp\_list:-

**Endpoint:** POST /get\_smtp\_list

**Auth Required:** Yes (uses req.authData.corporate\_id)

**Description:** Retrieves a paginated and optionally filtered/sorted list of SMTP access entries for a corporate user.

**Request Body:**

- pageno (required), perpage, sortbyfield, ascdesc, searchkey (optional)

**Search Filters:** Performs case-insensitive search on smtp\_name, username, from\_email\_address, coupon\_code, and status.

**Response:**

- status: 'success' with smtp\_list on success
- status: 'val\_err' or status: 'error' with a message on failure

**add\_smtp\_access:-**

**Endpoint:** POST /add\_smtp\_access

**Auth Required:** Yes (uses req.authData.corporate\_id)

**Description:** Adds a new SMTP access configuration for a corporate user.

**Request Body:**

- smtp\_name, host\_address, username, from\_email\_address, password, port (all required)
- method (required, must be "SSL" or "TLS")

**Response:**

- status: 'success' with created SMTP object on success
- status: 'val\_err' or status: 'error' with a message on failure

**update\_smtp\_setup\_data :-**

**Endpoint:** POST /update\_smtp\_setup\_data

**Auth Required:** Yes (uses req.authData.corporate\_id)

**Description:** Updates an existing SMTP access configuration based on its ID.

**Request Body:**

- smtp\_id, smtp\_name, host\_address, from\_email\_address, password, method ("SSL" or "TLS"), and port (all required)

**Functionality:** Updates the corresponding SMTP document with new values and sets status to "inactive" and updates the timestamp.

**Response:**

- status: 'success' with the updated record on success
- status: 'val\_err' or status: 'error' with validation or update failure messages

**update\_smtp\_setup\_status:-**

**Endpoint:** POST /update\_smtp\_setup\_status

**Auth Required:** Yes

**Description:** Updates the `status` (either `active` or `inactive`) of an existing SMTP configuration.

**Request Body:**

- `smtp_id` (required): ID of the SMTP configuration
- `status` (required): Must be either `"active"` or `"inactive"`

**Response:**

- `status`: `'success'` with confirmation message and update result on success
- `status`: `'val_err'` or `status`: `'error'` if validation fails or update encounters an error

`delete_smtp_setup`:-

**Endpoint:** `POST /delete_smtp_setup`

**Auth Required:** Yes

**Description:** Deletes an existing SMTP access configuration by its ID.

**Request Body:**

- `smtp_id` (required): The ID of the SMTP configuration to be deleted
- Functionality:** Validates input and removes the matching SMTP entry from the database.
- Response:**
- `status`: `'success'` with a confirmation message and deleted data
  - `status`: `'val_err'` or `status`: `'error'` on validation or deletion failure

`send_email`:-

**Function:** `send_email`

**Auth Required:** No (assumed internal utility function)

**Description:** Sends an email using SMTP configuration from the database or a default SMTP setup.

**Parameters:**

- `smtp_id` (optional): ID of SMTP config; uses default if not provided
- `to_mail`, `subject`, `email_for` (required): Recipient email, email subject, and template name or message
- `template_modal` (optional): Data to render the HTML template using Mustache
- `attachments` (optional): File attachments for the email

**Returns:**

- A result message from the `emailConfig.sendEmail` process
- Throws error if SMTP config is invalid or email sending fails