Vyaguta Leave Data API Documentation

Table of Contents

1. Introduction

The Vyaguta Leave Data API allows users to fetch, process, and visualize data related to employee leave from the Vyaguta application. The API is designed to automate data acquisition, ETL (Extract, Transform, Load) processes, and provide sample visualizations in a user-friendly manner.

This document provides detailed information on the available API endpoints, their parameters, usage examples, and response formats.

2. Authentication

All API requests require an `X-Custom-Passcode` header for authorization. The passcode ensures secure access to the API resources. Failure to provide the correct passcode will result in an authentication error.

Sample Header:

```
X-Custom-Passcode: passcode_here
```

3. API Endpoints

3.1 Fetching and Uploading Data

Triggers the function to acquire data from the Vyaguta application.

If `category` is set to `api` or not provided, the API defaults to using yesterday's date for `start_date` and today's date for `end_date`. If `category` is set to `csv`, the `csv_file` parameter is required.

URL: `/api/v1/acquire/insert`

Method: `POST`

Headers:

- `X-Custom-Passcode` : Required, string.

Parameters:

- `start_date`: Optional, string in `YYYY-MM-DD` format. Defaults to yesterday.
- `end_date`: Optional, string in `YYYY-MM-DD` format. Defaults to today.
- `csv_file`: Required when `category` is set to `csv`, file path to the CSV file for upload (e.g., `"/path/to/file"`).
- `category`: Required, string indicating the category type. Can be `csv` or `api`.

Response:

> ```
> {
>        "message": "Insert operation started for leave Table initiated as 2024-09-12"
>    }
> ```

Usage Examples:

Example 1: Upload CSV File

```
curl --location 'http://0.0.0.0:4448/api/v1/acquire/insert' \
   --header 'X-Custom-Passcode: ZG5OalFBPT0hISYhIU1qQXlNdz09' \
   --form 'csv_file=@"/path/to/file"' \
   --form 'category="csv"'
```

Example 2: Acquire Data via API

```
curl --location 'http://0.0.0.0:4448/api/v1/acquire/insert' \
    --header 'X-Custom-Passcode: ZG5OalFBPT0hISYhIU1qQXlNdz09' \
    --form 'start_date="2024-10-01"' \
    --form 'end_date="2024-10-02"' \
    --form 'category="api"'
```

Example 3: Acquire Data via API without Dates

```
curl --location 'http://0.0.0.0:4448/api/v1/acquire/insert' \
    --header 'X-Custom-Passcode: ZG5OalFBPT0hISYhIU1qQXlNdz09' \
    --form 'category="api"'
```

3.2 ETL Process

Triggers the ETL process for specific data sets.

If `inserted_date` is not provided, the current date is used. If `etl_name` is not specified, the API processes all ETL jobs including `user`, `leave`, `designation`and `leave_txn`.

URL: **`/api/v1/etl/load`**

Method: **`POST`**

Headers:

- `X-Custom-Passcode` : Required, string.

Parameters:

- `inserted_date`: Optional, string in `YYYY-MM-DD` format. Defaults to today.
- `etl_name`: Optional, string. If not provided, processes `user`, `leave`, `designation`,and `leave_txn`.

Response:

> ```
> {
> ```

```
        "message": "ETL process started for user Table as of 2024-09-07"
    }
```

Usage Example:

```
 curl --location 'http://0.0.0.0:4448/api/v1/etl/load' \
--header 'X-Custom-Passcode: passcode_here' \
--form 'inserted_date="2024-09-07"' \
--form 'etl_name="user"'
```

3.3 Sample Visualization

Provides a sample of the data in HTML format.

Returns an HTML page with sample data visualizations. The data is for sample purposes and can be viewed as a webpage.

URL:

- `/sample` - Returns a sample of the data without any filters.
- `/sample?startdate={startdate}&enddate={enddate}` - Returns a sample of the data filtered by the specified date range.

Method: `GET`

Query Parameters:

- `startdate` (optional) - The start date of the date range in the format YYYY-MM-DD.
- `enddate` (optional) - The end date of the date range in the format YYYY-MM-DD.

Headers:

- None

Response:

Returns an HTML page with embedded images of various visualizations based on the provided date range. If no date range is provided, the data is not filtered.

Usage Examples:

```
curl --location 'http://0.0.0.0:4448/sample'
```

```
curl --location 'http://0.0.0.0:4448/sample?startdate=2023-03-01&enddate=2024-07-01'
```

3.4 Visualize Download

Downloads a specific type of visualization.

Provides downloadable visualizations based on the specified plot type. Supported plot types include `department`, `supervisor`, `designation`, and `leave`.

URL: **`/api/v1/viz/sample/download/{plot_type}`**

Method: **`GET`**

Headers:

- `X-Custom-Passcode` : Required, string.

Parameters:

- `plot_type`: Required, string. The type of plot to download. Valid values are `department`, `supervisor`, `designation`, and `leave`.

Response:

> Returns a downloadable file containing the requested visualization.

Usage Example:

```
curl --location 'http://0.0.0.0:4448/api/v1/viz/sample/download/department' \
--header 'X-Custom-Passcode: passcode_here'
```

3.5 Leave Balance Data in JSON Format

Retrieves leave balance data for employees in JSON format for a given date range.

Use this API to get a structured summary of the leave data, which includes the available leave, leave taken, leave type, and total leave for each employee. The `start_date` and `end_date` parameters are used to filter the leave data based on the specified date range.

URL: `/api/v1/viz/leavebalance`

Method: `GET`

Headers:

- `X-Custom-Passcode` : Required, string. Authentication passcode for accessing the API.

Parameters:

- `start_date`: Required, string. The start date for fetching leave data in `YYYY-MM-DD` format (e.g., "2020-09-08").
- `end_date`: Required, string. The end date for fetching leave data in `YYYY-MM-DD` format (e.g., "2024-09-12").

Response:

```
{
        "Data": {
            "01011": [
                {
                    "available_leave": 6,
                    "leave_taken": 0,
                    "leave_type": "Menstruation",
                    "total_leave": 6
                },
                {
                    "available_leave": 2,
                    "leave_taken": 4,
                    "leave_type": "Sick",
                    "total_leave": 6
                }
            ],
            "02022": [
                {
                    "available_leave": 8,
                    "leave_taken": 2,
                    "leave_type": "Casual",
                    "total_leave": 10
                }
            ]
        }
    }
```

Usage Example:

```
curl --location --request GET 'http://0.0.0.0:4448/api/v1/viz/leavebalance' \
    --header 'X-Custom-Passcode: ZG5OalFBPT0hISYhIU1qQXlNdz09' \
    --form 'start_date="2020-09-08"' \
    --form 'end_date="2024-09-12"'
```

3.6 Leave Balance Analysis Report (PNG)

Retrieves a detailed visual leave balance analysis report in PNG format for the specified employees.

Use this API to get a visual summary of the leave balances for the employees. This endpoint is useful for quickly generating an overview image of leave statistics.

URL: **`/api/v1/viz/leavebalance/analysis`**

Method: **`GET`**

Headers:

- `X-Custom-Passcode` : Required, string. Authentication passcode for accessing the API.

Parameters:

- `emp_id`: Optional, string. Comma-separated list of employee IDs (e.g., "01011,02022"). If not provided, the report will be generated for all employees.

Response:

> ```
> Binary data representing the PNG report.
> ```

Usage Example:

```
curl --location 'http://0.0.0.0:4448/api/v1/viz/leavebalance/analysis' \
   --header 'X-Custom-Passcode: ZG5OalFBPT0hISYhIU1qQXlNdz09' \
   --form 'emp_id="01011,02022"'
```

3.7 Employee Details in JSON Format

Retrieves detailed employee information such as department, designation, email, employee ID, and role within the organization.

Use this API to get detailed employee data for a specified username. If `user_name` is not provided, the API will return details for all users.

URL: `/api/v1/viz/employee`

Method: `GET`

Headers:

- `X-Custom-Passcode` : Required, string. Authentication passcode for accessing the API.

Parameters:

- `user_name`: Optional, string. The username to filter employee details (e.g., "Bibek").

Response:

> ```
> {
>        "Data": [
>            {
>                "department_description": "Engineering",
>                "designation_name": "Software Engineer",
>                "email": "bibekshrestha@lftechnology.com",
>                "emp_id": "543422",
>                "full_name": "Bibek Shrestha",
>                "is_hr": 1,
> ```

```
            "is_supervisor": 0
        },
        {
            "department_description": "HR and Administration",
            "designation_name": "Software Engineer",
            "email": "bibekthapa@lftechnology.com",
            "emp_id": "1652",
            "full_name": "Bibek Thapa",
            "is_hr": 0,
            "is_supervisor": 0
        }
    ]
}
```

Usage Example:

```
curl --location --request GET 'http://0.0.0.0:4448/api/v1/viz/employee' \
   --header 'X-Custom-Passcode: ZG5OalFBPT0hISYhIU1qQXlNdz09' \
   --form 'user_name="Bibek"'
```

4. Response Formats

The API responses are generally in JSON format for data operations and downloadable files for visualizations. HTML format is used for sample visualization.