# Tables

configtable

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| automation\_id | INT | Primary key, unique identifier for automation |
| status | BOOLEAN | Indicates if the automation is active or disabled |
| default\_time | TIME | Default time of day for the automation to run |
| override\_time | TIME | Nullable, overridden time of day if specified |
| description | VARCHAR(255) | Text description of what the automation does |
| type | VARCHAR(255) | Type of the task, for categorization |

Apenas preciso de 4 records que serão atualizadas por um crm.

Toda noite. Vou fazer uma cron via springboot para atualizar para valores

pontotable

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |  |
| log\_id | INT | Primary key, unique identifier for the log entry |  |
| automation\_id | INT | Foreign key, references automation\_id in Automations table |  |
| timestamp | TIMESTAMP | Time and date when the automation was executed |  |
| status | VARCHAR(50) | Outcome of the automation (SUCCESS, FAILED, etc.) |  |
| message | TEXT | Detailed message or log output from the automation process |  |

Updated by python script

# Casos de uso

1. **Automations Table**

**Purpose**: To store and manage the configuration and status of each automation task.

**Use Cases**:

* **Retrieve Automation Settings**: The Python script queries this table to determine the current settings for each automation, such as whether it's enabled and the execution times (default or overridden).
* **Update Automation Settings**: The API updates this table in response to changes made through the mobile app, such as disabling an automation or changing its execution time.
* **Automation Scheduling**: The script uses the default\_time and override\_time to schedule the automation tasks accordingly.
* **View Current Automations**: Used by system administrators or via the mobile app to view the current configuration and status of all automation tasks.

### 2. ****Automation Logs Table****

**Purpose**: To record the execution details of each automation task, providing a history of when and how automations were performed.

**Use Cases**:

* **Error Tracking and Debugging**: After an automation runs, the script logs the outcome. These logs are crucial for debugging issues if an automation fails.
* **Audit and Compliance**: Maintaining a detailed log of automation activities can be useful for audit trails and compliance with internal or external regulations.
* **Performance Monitoring**: Analysis of the logs can help in identifying performance trends or recurring issues with certain automations, facilitating optimizations.

# Endpoints

* POST "http://localhost:8080/api/automation/change/time1?overrideTime=14:00:00"

Esse endpoint serve para setar o overridetime para algum valor

* POST <http://localhost:8080/api/automation/disable/time1>

Esse endpoint serve para desabilitar uma das batidas de ponto.

A escolha são: ‘time1’ , ‘time2’, ‘time3’,’time4’

* POST [http://localhost:8080/api/automation/disable-](http://localhost:8080/api/automation/disable/time1)all

Esse endpoint desativa todos os pontos do dia

* GET [http://localhost:8080/api/automation/settings](http://localhost:8080/api/automation/disable/time1)
* resposta

[

    {

        "id": 1,

        "status": true,

        "defaultTime": "11:20:00",

        "overrideTime": null,

        "type": "time1"

    },

...

]

Esse endpoint volta com informações sobre a base de dados