1. Employee data

|  |  |  |
| --- | --- | --- |
| Key | Type | Description |
| id | Required | Unique employee identifier in the format ‘UIXXXXXXX’ where the X is replaced with alphanumeric |
| name | Required | Name of the employee |
| email\_address | Required | Email address of the employee. Follows the typical email address format. |
| phone\_number | Required | Phone number of the employee. Starts with either 9 or 8 and have 8 digits. |
| gender | Required | Gender of the employee (Male/Female) |

1. Café data

|  |  |  |
| --- | --- | --- |
| Key | Type | Description |
| name | Required | Name of the cafe |
| description | Required | A short description of the cafe |
| logo | ***Optional to implement*** | Logo of the café. This will be used to display a logo image on the front-end. |
| location | Required | Location of the cafe |
| id | Required | UUID |

1. Which employee work for which café, and the employee start date
2. No same employee can work in 2 cafes (this constraint can be handled either within the code or database)

* Create a GET endpoint /cafes?location=<location>

The response of this endpoint should be the below and sorted by the highest number of employees first

If a valid location is provided, it will filter the list to return only cafes that is within the area

If an invalid location is provided, it should return an empty list

If no location is provided, it should list down all cafes

|  |  |
| --- | --- |
| Key | Description |
| name | Name of the cafe |
| description | A short description of the cafe |
| employees | Number of the employees.  It must be an integer |
| logo *(optional)* | Logo of the café. This will be used to display a logo image on the front-end. |
| location | Location of the cafe |
| id | UUID |

* Create a GET endpoint /employees?cafe=<café>

The response of this endpoint should be the below and sorted by the highest number of days worked. It should list all the employees.

If a café is provided, it should list down only employees that belong to that café.

|  |  |
| --- | --- |
| Key | Description |
| id | Unique employee identifier in the format ‘UIXXXXXXX’ where the X is replaced with alpha numeric |
| name | Name of the employee |
| email\_address | Email address of the employee. |
| phone\_number | Phone number of the employee. |
| email\_address | Email address of the employee. |
| days\_worked | Number of days the employee worked  It must be an integer and is derived from the current date minus the start date of the employee in the cafe |
| cafe | Café’s name that the employee is under [leave blank if not assigned yet] |

* Create a POST endpoint /cafe

This should create a new café in the database.

* Create a POST endpoint /employee

This should create a new employee in the database.

This should also create the relationship between an employee and a café.

* Create a PUT endpoint /cafe

This should update the details of an existing café in the database.

* Create a PUT endpoint /employee

This should update the details of an existing employee in the database.

This should also update the relationship between an existing employee and a café.

* Create a DELETE endpoint /cafe

This should delete an existing café in the database. It should also delete all employees under the deleted cafe

* Create a DELETE endpoint /employee

This should delete an existing employee in the database.