

Final Project: API Spec

Delivered by Julian Peterson

ONID: peterju4

CS 493: Cloud Application Development

Fall 2023

Oregon State University

Last Update: Dec 10. 1:45 pm Pacific

APP URL: <https://final-project-407721.wl.r.appspot.com>

Change log	1
Data Model	2
Create a Restaurant	3
Get a Restaurant	6
List all Restaurants	8
Edit a Restaurant - PUT	10
Edit a Restaurant - PATCH	13
Delete a Restaurant	16
Create an Employee	18
Get an Employee	21
List all Employees	23
Edit an Employee - PUT	25
Edit an Employee - PATCH	27
Delete an Employee	29
Restaurant hires an Employee	31
Restaurant fires an Employee/Employee leaves a restaurant	33
List all Users	35

Change log

Version	Change	Date
1.0	Initial version.	Oct 6, 2022
2.0	Version updated for Final Project Submission	Dec 10, 2023

Data Model

The app stores three kinds of entities in Datastore: Users, Restaurants and Employees.

Users (can be understood as “owners” or “restaurant owners”)

Property	Data Type	Notes
id	String	The unique id of the user. It comes from the “sub” value of a JWT. The app automatically generates it and saves it upon login of a new user.

Restaurants

Property	Data Type	Notes
id	Integer	The id of the restaurant. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	string	The name of the restaurant.
cost	string	The average cost of dining at this restaurant given the following scale: \$: 10-20 \$\$: 20-50 \$\$\$: 50 - 100 \$\$\$\$: 100+
cuisine	string	The cuisine of food in which the restaurant specializes
employees	list	A list of employees working at the restaurant. Each employee is represented by a JSON object with three attributes: id, name, and a self url.
owner	String	The unique id of the owner (user) of the restaurant.
self	string (link)	The URL to the endpoint to see more information about the restaurant.

Employees

Property	Data Type	Notes
id	Integer	The id of the employee. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	First and last name of the employee (“First Last”).
wage	Float	The hourly wage of the employee.
position	String	The role or position of the employee.
workplace	JSON object	A JSON object representing the employee's place of work. The JSON contains three attributes for the restaurant: id, name, and a self url.
self	String (link)	The URL to the endpoint to see more information about the employee.

Create a Restaurant

Allows you to create a new restaurant.

POST /restaurants

Request

Path Parameters

None

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the restaurant.	Yes
cost	The average cost of dining at this restaurant given the following scale: \$: 10-20 \$\$: 20-50 \$\$\$: 50 - 100 \$\$\$\$: 100+	Yes
cuisine	Type of food served at the restaurant.	Yes

Request Body Example

```
{
  "name": "Ivan's",
  "cost": "$$$",
  "cuisine": "Italian"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the restaurant will not be created, and 400 status code will be returned.

		<p>If the request includes extraneous attributes (i.e., location), the restaurant will not be created, and a 400 status code will be returned.</p> <p>Content validation occurs for the value of the “cost” attribute. A 400 status code will be returned if the “cost” value is not “\$”, “\$\$”, “\$\$\$”, or “\$\$\$\$”.</p>
Failure	401 Unauthorized	If the request is sent without a valid bearer token associated with a valid JWT and account, then the restaurant will not be created, and a 401 status code will be returned.
Failure	406 Not Acceptable	If the request’s Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. The app will send back this value in the response body as shown in the example.
- The value of the attribute `employees` is a list representing the employees who work at the given. The list contains JSON objects, each representing an employee with the following attributes: id, name, and a self url. Upon creation, and if the restaurant has no employees, the value of `employees` should be an empty list.

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Ivan's",
  "cost": "$$$",
  "cuisine": "Italian",
  "employees": [],
  "owner": 9403762038846693,
  "self": "localhost:8080/restaurants/123"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The request object includes extraneous attributes"
}
```

Status: 400 Bad Request

```
{  
  "Error": "The 'cost' value must equal $, $$, $$$, or $$$$"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "Invalid token"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "Missing token"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

Get a Restaurant

Allows you to get an existing restaurant

GET /restaurants/:restaurant_id

Request

Path Parameters

Name	Description
restaurant_id	ID of the restaurant

Request Body

None

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid or missing authorization header
Failure	403 Forbidden	Valid token, but not to view this restaurant
Failure	404 Not Found	No restaurant with this restaurant_id exists
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Ivan's",
  "cost": "$$$",
  "cuisine": "Italian",
  "employees": [],
  "owner": 9403762038846693,
  "self": "localhost:8080/restaurants/123"
}
```

Failure

Status: 401 Unauthorized

```
{  
  "Error": "Invalid token"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "You are not authorized to view this restaurant"  
}
```

Status: 404 Not Found

```
{  
  "Error": "No restaurant with this restaurant_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

List all Restaurants

List all the restaurants owned by the authorized user.

GET /restaurants

Request

Path Parameters

None

Request Body

None

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid or no token
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
  "count": 12,
  "restaurants": [
    {
      "id": 123,
      "name": "Ivan's",
      "cost": "$$$",
      "cuisine": "Italian",
      "employees": [],
      "owner": 9403762038846693,
      "self": "localhost:8080/restaurants/123"
    },
    {
      "id": 456,
      "name": "WhistlePig",
      "cost": "$",
```



```

    "cuisine": "Coffee",
    "employees": [],
    "owner": 9403762038846693,
    "self": "localhost:8080/restaurants/456"
  },
  {
    "id": 789,
    "name": "Indulge",
    "cost": "$$$$",
    "cuisine": "Thai",
    "employees": [],
    "owner": 9403762038846693,
    "self": "localhost:8080/restaurants/789"
  },
  {
    "id": 735,
    "name": "Nancy's",
    "cost": "$$",
    "cuisine": "Brunch",
    "employees": [],
    "owner": 9403762038846693,
    "self": "localhost:8080/restaurants/735"
  },
  {
    "id": 194,
    "name": "Social Butterfly",
    "cost": "$",
    "cuisine": "Shawarma",
    "employees": [],
    "owner": 9403762038846693,
    "self": "localhost:8080/restaurants/194"
  }
],
"next": "localhost:8080/restaurants?limit=5?offset=5"
}

```

Failure

Status: 401 Unauthorized

```

{
  "Error": "Invalid token"
}

```

Status: 406 Not Acceptable

```

{
  "Error": "This endpoint only supports the return of JSON objects"
}

```

Edit a Restaurant - PUT

Allows you to edit all attributes of a restaurant.

PUT /restaurants/:restaurant_id

Request

Path Parameters

Name	Description
restaurant_id	ID of the restaurant

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the restaurant.	Yes
cost	The average cost of dining at this restaurant given the following scale: \$: 10-20 \$\$: 20-50 \$\$\$: 50 - 100 \$\$\$\$: 100+	Yes
cuisine	Type of food served at the restaurant.	Yes

Request Body Example

```
{
  "name": "Ivan's",
  "cost": "$$",
  "cuisine": "Italian-Fusion"
}
```

Response

Response Body Format

Success: No Body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the restaurant will not be updated, and a 400 status code will

		<p>be returned.</p> <p>If the request includes extraneous attributes (i.e., location), the restaurant will not be updated, and a 400 status code will be returned.</p> <p>Content validation occurs for the value of the “cost” attribute. A 400 status code will be returned if the “cost” value is not “\$”, “\$\$”, “\$\$\$”, or “\$\$\$\$”.</p> <p># note - when creating this, just make a list with these strings in it. use: if content[“cost”] not in cost_list: return 400</p>
Failure	401 Unauthorized	If the request is sent without a valid bearer token associated with a valid JWT and account, then the restaurant will not be updated, and a 401 status code will be returned.
Failure	403 Forbidden	Valid token, but not to edit this restaurant
Failure	404 Not Found	No restaurant with this restaurant_id exists
Failure	406 Not Acceptable	If the request’s Accept header does not include JSON, the restaurant will not be updated, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The request object includes extraneous attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The 'cost' value must equal $, $$, $$$, or $$$$"
}
```

Status: 401 Unauthorized

```
{
  "Error": "Invalid token"
}
```

```
}
```

Status: 403 Forbidden

```
{  
  "Error": "You are not authorized to view this restaurant"  
}
```

Status: 404 Not Found

```
{  
  "Error": "No restaurant with this restaurant_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

Edit a Restaurant - PATCH

Allows you to edit zero, one, or two attributes of a restaurant.

PATCH /restaurants/:restaurant_id

Request

Path Parameters

Name	Description
restaurant_id	ID of the restaurant

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the restaurant.	No
cost	The average cost of dining at this restaurant given the following scale: \$: 10-20 \$\$: 20-50 \$\$\$: 50 - 100 \$\$\$\$: 100+	No
cuisine	Type of food served at the restaurant.	No

Request Body Example

```
{
  "name": "Ivan's Trattoria"
}
```

Response

Response Body Format

Success: No Body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	If the request includes all 3 attributes, the restaurant will not be updated, and a 400 status code will be returned. This type of update should use a PUT request rather than a

		<p>PATCH request.</p> <p>If the request includes extraneous attributes (i.e., location), the restaurant will not be updated, and a 400 status code will be returned.</p> <p>Content validation occurs for the value of the “cost” attribute. A 400 status code will be returned if the “cost” value is not “\$”, “\$\$”, “\$\$\$”, or “\$\$\$\$”.</p>
Failure	401 Unauthorized	If the request is sent without a valid bearer token associated with a valid JWT and account, then the restaurant will not be updated, and a 401 status code will be returned.
Failure	403 Forbidden	Valid token, but not to edit this restaurant
Failure	404 Not Found	No restaurant with this restaurant_id exists
Failure	406 Not Acceptable	If the request’s Accept header does not include JSON, the restaurant will not be updated, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object includes all three attributes. Please use a PUT request
to update all three attributes of a restaurant"
}
```

Status: 400 Bad Request

```
{
  "Error": "The request object includes extraneous attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The 'cost' value must equal $, $$, $$$, or $$$$"
}
```

Status: 401 Unauthorized

```
{
  "Error": "Invalid token"
}
```

Status: 403 Forbidden

```
{  
  "Error": "You are not authorized to view this restaurant"  
}
```

Status: 404 Not Found

```
{  
  "Error": "No restaurant with this restaurant_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

Delete a Restaurant

Allows you to delete a restaurant. Note that if the restaurant has employees working there, deleting the restaurant updates the workplace of all those employees equal to None.

DELETE /restaurants/:restaurant_id

Request

Path Parameters

Name	Description
restaurant_id	ID of the restaurant

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If the request is sent without a valid bearer token associated with a valid JWT and account, then the restaurant will not be deleted, and a 401 status code will be returned.
Failure	403 Forbidden	Valid token, but not to delete this restaurant
Failure	404 Not Found	No restaurant with this restaurant_id exists
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be deleted, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
```



```
"Error": "Invalid token"
}
```

Status: 403 Forbidden

```
{
  "Error": "You are not authorized to view this restaurant"
}
```

Status: 404 Not Found

```
{
  "Error": "No restaurant with this restaurant_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint only supports the return of JSON objects"
}
```

Create an Employee

Allows you to create a new employee.

POST /employees

Request

Path Parameters

None

Request Authorization

Not required

Authorization Type

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
Name	First and last name of the employee ("First Last").	Yes
Wage	The hourly wage of the employee.	Yes
Position	The role or position of the employee.	Yes

Request Body Example

```
{
  "name": "Tim Reilly",
  "wage": 19.50,
  "position": "Bartender"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	<p>If the request is missing any of the 3 required attributes, the employee will not be created, and 400 status code will be returned.</p> <p>If the request includes extraneous attributes (i.e., height), the employee will not be created, and a 400 status code will be returned.</p>

		Content validation occurs for the value of the “wage” attribute. A 400 status code will be returned if the “wage” value is less than 14.20, the minimum wage in Oregon.
Failure	406 Not Acceptable	If the request’s Accept header does not include JSON, the employee will not be created, and a 406 status code will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value will be sent in the response body as shown in the example.
- The value of the attribute `workplace` is a JSON representing the restaurant at which the employee currently works. The JSON contains three attributes for the restaurant: `id`, `name`, and a self url. Upon creation, and if the employee isn’t employed, the value of `workplace` should be null.

Success

Status: 201 Created

```
{
  "id": 6666,
  "name": "Tim Reilly",
  "wage": 19.50,
  "position": "Bartender",
  "workplace": null,
  "self": "localhost:8080/employees/6666"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The request object includes extraneous attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The 'wage' value must not be less than 14.20"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint only supports the return of JSON objects"
}
```

}

Get an Employee

Allows you to get an existing employee.

GET /employees/:employee_id

Request

Path Parameters

Name	Description
employee_id	ID of the employee

Request Authorization

Not required

Authorization Type

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No employee with this employee_id exists
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
  "id": 6666,
  "name": "Tim Reilly",
  "wage": 19.50,
  "position": "Bartender",
  "workplace": null,
  "self": "localhost:8080/employees/6666"
}
```

Failure

Status: 404 Not Found

```
{
  "Error": "No employee with this employee_id exists"
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

List all Employees

List all the employees.

GET /employees

Request

Path Parameters

None

Request Authorization

Not required

Authorization Type

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
  "count": 22,
  "employees": [
    {
      "id": 6666,
      "name": "Tim Reilly",
      "wage": 19.50,
      "position": "Bartender",
      "workplace": null,
      "self": "localhost:8080/employees/6666"
    },
    {
      "id": 1973,
      "name": "Furhad Sultani",
      "wage": 23.19,
      "position": "Chef",
      "workplace": null,
    }
  ]
}
```

```

    "self": "localhost:8080/employees/1973"
  },
  {
    "id": 3999,
    "name": "Sofia Bartel",
    "wage": 21.12,
    "position": "Hostess",
    "workplace": null,
    "self": "localhost:8080/employees/3999"
  },
  {
    "id": 5520,
    "name": "Jacob Pressly",
    "wage": 15.50,
    "position": "Server",
    "workplace": null,
    "self": "localhost:8080/employees/5520"
  },
  {
    "id": 3098,
    "name": "Gim Shadowbear",
    "wage": 18.00,
    "position": "Line Cook",
    "workplace": null,
    "self": "localhost:8080/employees/3098"
  }
],
"next": "localhost:8080/employees?limit=5?offset=5"
}

```

Failure

Status: 406 Not Acceptable

```

{
  "Error": "This endpoint only supports the return of JSON objects"
}

```


Edit an Employee - PUT

Allows you to edit all attributes of an employee.

PUT /employee/:employee_id

Request

Path Parameters

Name	Description
employee_id	ID of the employee

Request Authorization

Not required

Authorization Type

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
Name	First and last name of the employee ("First Last").	Yes
Wage	The hourly wage of the employee.	Yes
Position	The role or position of the employee.	Yes

Request Body Example

```
{
  "name": "Tim Reilly",
  "wage": 25.50,
  "position": "Lead Bartender"
}
```

Response

Response Body Format

Success: No Body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	<p>If the request is missing any of the 3 required attributes, the employee will not be created, and 400 status code will be returned.</p> <p>If the request includes extraneous attributes (i.e., height), the employee will not be created, and a 400 status code</p>

		will be returned. Content validation occurs for the value of the “wage” attribute. A 400 status code will be returned if the “wage” value is less than 14.20, the minimum wage in Oregon.
Failure	404 Not Found	No employee with this employee_id exists
Failure	406 Not Acceptable	If the request’s Accept header does not include JSON, the restaurant will not be updated, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The request object includes extraneous attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The 'wage' value must not be less than 14.20"
}
```

Status: 404 Not Found

```
{
  "Error": "No employee with this employee_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint only supports the return of JSON objects"
}
```

Edit an employee - PATCH

Allows you to edit zero, one, or two attributes of an employee.

PATCH /employees/:employee_id

Request

Path Parameters

Name	Description
employee_id	ID of the employee

Request Authorization

Not required

Authorization Type

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
Name	First and last name of the employee ("First Last").	No
Wage	The hourly wage of the employee.	No
Position	The role or position of the employee.	No

Request Body Example

```
{
  "wage": 30.00,
  "position": "Bar Manager"
}
```

Response

Response Body Format

Success: No Body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	<p>If the request includes all 3 attributes, the restaurant will not be updated, and a 400 status code will be returned. This type of update should use a PUT request rather than a PATCH request.</p> <p>If the request includes extraneous attributes (i.e., height),</p>

		<p>the employee will not be updated, and a 400 status code will be returned.</p> <p>Content validation occurs for the value of the “wage” attribute. A 400 status code will be returned if the “wage” value is less than 14.20, the minimum wage in Oregon.</p>
Failure	404 Not Found	No employee with this employee_id exists
Failure	406 Not Acceptable	If the request’s Accept header does not include JSON, the restaurant will not be updated, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object includes all three attributes. Please use a PUT request
to update all three attributes of an employee"
}
```

Status: 400 Bad Request

```
{
  "Error": "The request object includes extraneous attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The 'wage' value must not be less than 14.20"
}
```

Status: 404 Not Found

```
{
  "Error": "No employee with this employee_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "This endpoint only supports the return of JSON objects"
}
```

Delete an Employee

Allows you to delete an employee. If an employee is deleted, then the employing restaurant's employee list will be updated.

DELETE /employee/:employee_id

Request

Path Parameters

Name	Description
employee_id	ID of the employee

Request Authorization

Not required

Authorization Type

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No employee with this employee_id exists
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
  "Error": "No employee with this employee_id exists"
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

Restaurant hires an Employee

An employee begins work at a restaurant.

PUT /restaurants/:restaurant_id/employees/:employee_id

Request

Path Parameters

Name	Description
restaurant_id	ID of the restaurant
employee_id	ID of the employee

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Request Body

None

Note: Set Content-Length to 0 in your request when calling out to this endpoint. # not sure what this is about tbh

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a restaurant exists with this restaurant_id, an employee exists with this employee_id and the employee is unemployed (employee["workplace"] = null).
Failure	401 Unauthorized	Invalid token
Failure	403 Forbidden	The employee cannot work at this restaurant because they already have a workplace.
Failure	403 Forbidden	Valid token, but not to edit this restaurant
Failure	404 Not Found	The specified restaurant and/or employee does not exist.
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{  
  "Error": "Invalid token"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "You are not authorized to view this restaurant"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "The employee already has a workplace"  
}
```

Status: 404 Not Found

```
{  
  "Error": "The specified restaurant and/or employee does not exist"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

Comment

- An employee cannot have multiple workplaces.

Restaurant fires an Employee/Employee quits a Restaurant

Employee stops working at a restaurant. The Employee no longer has a workplace and the restaurant's employees decrease by 1.

DELETE /restaurants/:restaurant_id/employees/:employee_id

Request

Path Parameters

Name	Description
restaurant_id	ID of the restaurant
employee_id	ID of the employee

Request Authorization

Required

Authorization Type

Bearer Token (taken from JWT upon login)

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a restaurant exists with this restaurant_id, an employee exists with this employee_id and this employee works at this restaurant.
Failure	401 Unauthorized	Invalid token
Failure	403 Forbidden	Valid token, but not to edit this restaurant
Failure	404 Not Found	No employee with this employee_id works at the restaurant with this restaurant_id. This could be because no employee with this employee_id exists, or because no restaurant with restaurant_id exists, or even if both employee_id and restaurant_id are valid, the employee with this employee_id does not work at the restaurant with this restaurant_id.
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, the restaurant will not be created, and a 406 status code will be returned.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{  
  "Error": "Invalid token"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "You are not authorized to view this restaurant"  
}
```

Status: 404 Not Found

```
{  
  "Error": "No employee with this employee_id works at the restaurant with this  
restaurant_id"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```

List all Users

List all the restaurant owners.

GET /users

Request

Path Parameters

None

Request Authorization

Not required

Authorization Type

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the request's Accept header does not include JSON, a 406 status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
  "users": [
    {
      "id": 299236460109283601713,
      "restaurants": [
        {
          "id": 456,
          "name": "WhistlePig",
          "self": "localhost:8080/restaurants/456"
        },
        {
          "id": 123,
          "name": "Ivan's",
          "self": "localhost:8080/restaurants/123"
        }
      ]
    }
  ],
}
```

```
    "id": 388474553827372638479,  
    "restaurants": [  
      {  
        "id": 789,  
        "name": "Indulge",  
        "self": "localhost:8080/restaurants/789"  
      }  
    ]  
  }  
]
```

Failure

Status: 406 Not Acceptable

```
{  
  "Error": "This endpoint only supports the return of JSON objects"  
}
```