Restify API Documentation

Project Structure

Project Name: restify

Apps: accounts, blogs, notifications, restaurants.

Structure:

1. **accounts**
    - **migrations**
        - **contains migration files for models declared in this app**
    - **views**
        - **edit.py**
            - **contains view for user to edit their account**
        - **signup.py**
            - **contains view for client to register and account**
    - **__init__.py**
    - **admin.py**
            - **contains registration models in this app**
    - **apps.py**
            - **configurates app**
    - **models.py**
            - **contains models' declaration for this app**
            - **Models: User, Follow, LikesRestaurant, Comments, LikesBlog**
    - **permission.py**
            - **contains custom permission associated with app views**
    - **serializer.py**
        - **contains custom serializer need for account views**
    - **test.py**

        - Not used
    - urls.py
        - contains URL’s associated view account views

2. blogs
    - migrations
        - contains migration files for models declared in this app
    - views
        - feed.py
            - contains view for user to retrieve list of blogs of restaurants they follow.
        - addBlog.py
            - contains view for user to add a blog to their restaurant
        - detailBlog.py
            - contains view for user retrieve and update their blog post
        - listBlog.py
            - contains view for user to retrieve list of blogs.
    - __init__.py
    - admin.py
            - contains registration models in this app
    - apps.py
            - configurates app
    - models.py
            - contains models’ declaration for this app
            - Models: Blog
    - permission.py
            - contains custom permission associated with app views
    - serializer.py
        - contains custom serializer need for account views
    - test.py
        - Not used
    - urls.py
        - contains URL’s associated view this apps views

3. media
    - blog_images
        - stores blog post images in the database
    - restaurant_images
        - stores restaurant images in the database
    - restaurant_logos
        - stores restaurant logos in the database
    - restaurant_menu
        - stores restaurant menu images in the database
    - user_avatar
        - stores user avatar in the database

4. notifications
    - migrations
        - contains migration files for models declared in this app
    - views
        - view.py
            - contains view for user retrieve a list of their notifications
    - __init__.py
    - admin.py
            - contains registration models in this app
    - apps.py
            - configurates app
    - models.py
            - contains models’ declaration for this app
            - Models: Notification
    - permission.py
            - contains custom permission associated with app views
    - serializer.py
        - contains custom serializer need for account views
    - test.py
        - Not used
    - urls.py
        - contains URL’s associated view this apps views

5. restaurants.
    - migrations
        - contains migration files for models declared in this app
    - views
        - generalinfo
            - edit.py
                - contain view to update restaurant general info
            - info.py
                - contains view to retrieve restaurant info
        - images
            - add.py
                - contains view to add image to restaurant
            - remove.py
                - contains view to remove image from restaurant
        - menu
            - add.py
                - contains view to add menu item to restaurant
            - update.py
                - contains view to update menu item in restaurant
        - create.py
                - contains view to create a restaurant
        - search,py
                - contains view to search for a restaurant
    - __init__.py

    - admin.py
            - contains registration models in this app
    - apps.py
            - configurates app
    - models.py
            - contains models' declaration for this app
            - Models: Restaurant, Image, Menu
    - permission.py
            - contains custom permission associated with app views
    - serializer.py
        - contains custom serializer need for account views
    - test.py
        - Not used
    - urls.py

        - contains URL's associated view this apps views

6. restify
    - __init__.py
    - asgi.py.py
    - settings.py
        - contains setting required to use the app
    - urls.py
        - contains urls which connect all apps with the project
    - wsgi.py
7. venv
    - contains packages for virtual environment
8. db.sqlite3
    - sqlite3 database for the project
9. .gitignore
    - Contains list of folders/files to ignore while uploading to git repository
10. manage.py
    - contain code to start the server when called
11. requirements.txt
    - contains list of required packages to run project
12. run.sh
    - contains script to activate virtual environment and start the server
13. startup.sh
    - contains script to run all preparation needed for our code to run in a new environment. It creates the virtual environment, install all required packages with pip, and run all migrations

Apps

accounts:

Models found in this app include:

- **User which extends djangos AbstractUser model**
- **Follows to connect users and the restaurants they follow**
- **LikesRestaurant to connect users and the restaurants they like**
- **Comments to connect comments to a restaurant with the user who commented**
- **LikesBlog to connect users and the blogs they like**

Views:

- **for signing up SignUpView**
- **for login TokenObtainPairView**
- **for editing profile EditView**

Custom Permissions:

- **IsOwner which checks if the authenticated user is an owner of the specified object**

blogs:

Models found in this app include:

- **Blog which holds information required to display a blog posted by a restaurant**

Views:

- **for creating blogs, AddBlogView**
- **for displaying individual blogs with a blog id parameter, DetailBlogView**
- **for displaying all blogs, ListBlogView**
- **for individual user feeds with a user id parameter, UserFeedView**
- **for the user action of liking a blog, LikeBlogView**
- **for the user action of unliking a blog, UnlikeBlogView**

Custom Permissions:

- IsOwnerOrReadOnly which checks if the user is the restaurant owner, if not allows read-only requests
- HasLikedBlog which checks if the user has already liked this blog
- HasNotLikeBlog which checks if the user has not yet liked this blog

media:

In the media app, subdirectories store organized images.

notifications:

Models found in this app include:

- Notification which holds information required to send notifications to specified users

In this notifications app, two views exist

- for the user specific viewing of Notifications, NotificationsView which lists all notifications for the authenticated user
- for the creation of Notifications, AddNotificationView which is used to send specified notifications to specified users

Custom Permissions:

- NotOwnOneRestaurant which restricts users from owning more than one restaurant
- CheckUserOwnsObj which checks if the authenticated user is an owner of the specified restaurant
- IsOwner which checks if the user is an owner of the specified object

restaurants:

Models found in this app include:

- Restaurant which holds information required to host a restaurant and its information
- Image which connects an image to a restaurant for display

- Menu which holds information required to display a menu item for a restaurant

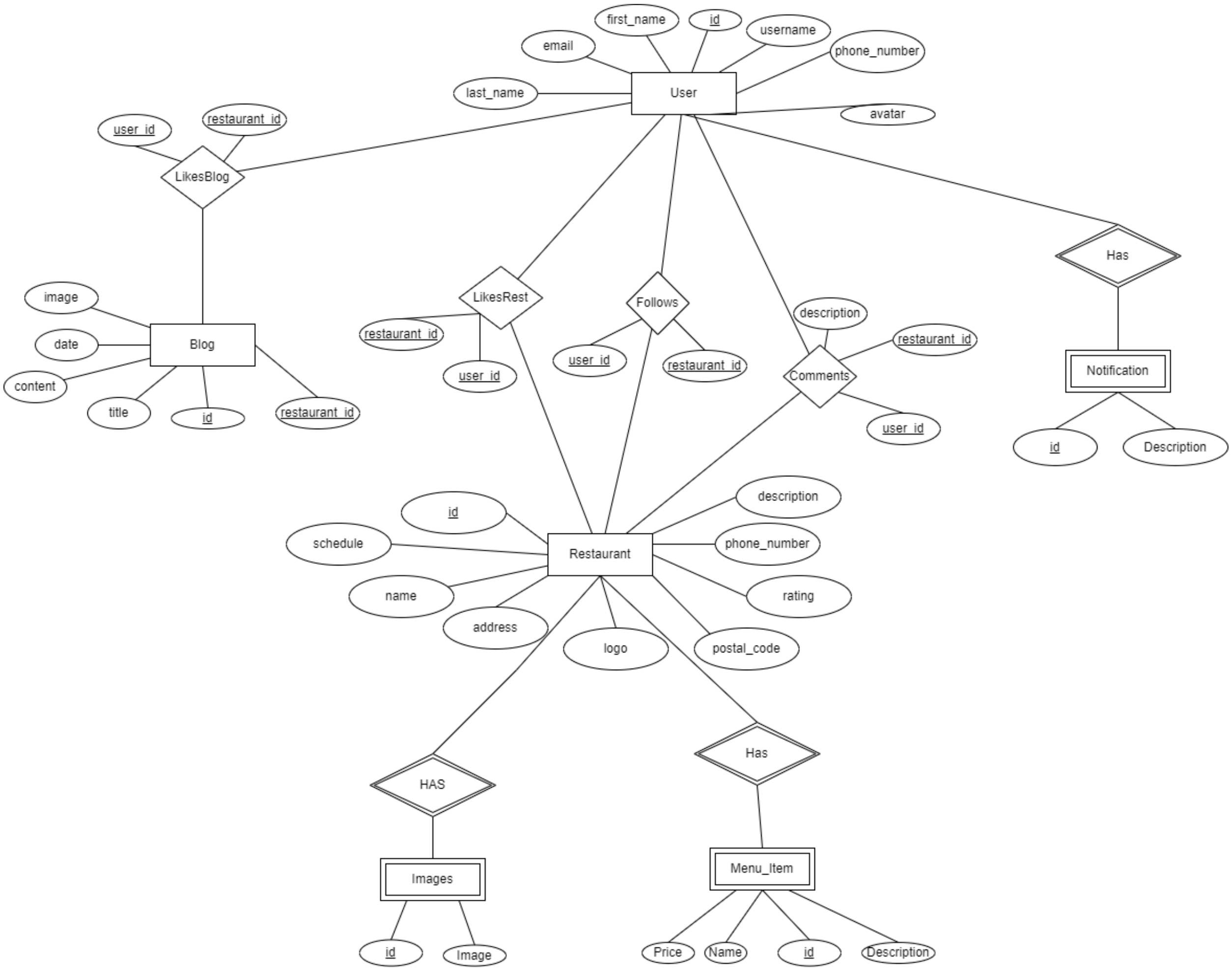
Views:

- for the creation of a restaurant, CreateRestaurantView
- for the searching of a restaurant based on the restaurants name, address, and menu items, SearchView
- for the editing of a restaurants information, UpdateInfoView
- for the addition of a menu item, AddMenuItemView
- for the editing of a menu item, UpdateMenuView
- for the addition of a restaurant image, AddImageView
- for the removal of a restaurant image, RemoveImageView
- for the action of liking a restaurant, LikeRestaurantView
- for the action of unliking a restaurant, UnlikeRestaurantView
- for the action of following a restaurant, FollowRestaurantView
- for the action of unfollowing a restaurant, UnfollowingRestaurantView
- for the action of commenting on a restaurant, CommentsRestaurantView

Custom Permissions:

- NotOwnOneRestaurant which restricts users from owning more than one restaurant
- CheckUserOwnsObj which checks if the user is an owner of the specified restaurant
- IsOwner which checks if the user is an owner of the specified object
- HasLikedRestaurant which checks if the user has already liked this restaurant
- HasNotLikedRestaurant which checks if the user has not yet liked this restaurant
- HasFollowedRestaurant which checks if the user has already followed this restaurant
- HasNotFollowedRestaurant which checks if the user has not yet followed this restaurant

Models


first_name
id
email
username
phone_number
last_name
User
avatar
user_id
restaurant_id
LikesBlog
Has
image
LikesRest
Follows
description
restaurant_id
restaurant_id
date
Blog
user_id
user_id
restaurant_id
Comments
Notification
content
user_id
title
id
restaurant_id
id
Description
description
id
schedule
Restaurant
phone_number
name
rating
address
logo
postal_code
Has
HAS
Images
Menu_Item
id
Image
Price
Name
id
Description

http://127.0.0.1:8000/ Explore

Restify API Documentation v1

[Base URL: 127.0.0.1:8000/]

http://127.0.0.1:8000/?format=openapi

Terms of service

Contact the developer

BSD License

Schemes

HTTP

Django Login Authorize

Filter by tag

accounts

GET /accounts/edit/{id}/ accounts_edit_read

Retrieve user profile of account with given id in url if client is owner of the acount and authenticated.

Parameters

Try it out

Name	Description
id * required integer *(path)*	A unique integer value identifying this user. id - A unique integer value identifying this use

Responses

Response content type application/json

Code	Description

Code	Description
200	Example Value **Model**

```
Edit    {
    email                   string($email)
                            title: Email address
                            maxLength: 254
    first_name              string
                            title: First name
                            maxLength: 150
    last_name               string
                            title: Last name
                            maxLength: 150
    phone_number*           string
                            title: Phone number
                            maxLength: 12
                            minLength: 1
    avatar                  string($uri)
                            title: Avatar
                            readOnly: true

}
```

PUT `/accounts/edit/{id}/` `accounts_edit_update`

Edits user profile of account with given id in url if client is owner of the acount and authenticated.

Parameters

Try it out

Name	Description
data * required `object` *(body)*	Example Value **Model**

```
Edit    {
    email                   string($email)
                            title: Email address
                            maxLength: 254
    first_name              string
                            title: First name
                            maxLength: 150
    last_name               string
                            title: Last name
                            maxLength: 150
    phone_number*           string
                            title: Phone number
                            maxLength: 12
                            minLength: 1

}
```

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this user. id - A unique integer value identifying this use

Responses Response content type application/json

Code	Description
200	Example Value **Model**

```
Edit  {
    email                    string($email)
                             title: Email address
                             maxLength: 254
    first_name               string
                             title: First name
                             maxLength: 150
    last_name                string
                             title: Last name
                             maxLength: 150
    phone_number*            string
                             title: Phone number
                             maxLength: 12
                             minLength: 1
    avatar                   string($uri)
                             title: Avatar
                             readOnly: true

}
```

PATCH `/accounts/edit/{id}/` `accounts_edit_partial_update`

Edits user profile of account with given id in url with given fields if client is owner of the acount and authenticated.

Parameters Try it out

Name	Description
data * required `object` *(body)*	Example Value **Model**

```
Edit  {
    email                    string($email)
                             title: Email address
                             maxLength: 254
    first_name               string
                             title: First name
                             maxLength: 150
    last_name                string
                             title: Last name
                             maxLength: 150
    phone_number*            string
                             title: Phone number
                             maxLength: 12
                             minLength: 1

}
```

id * required `integer` *(path)*	A unique integer value identifying this user. id - A unique integer value identifying this use

Responses

Response content type application/json

Code	Description
200	Example Value **Model**

```
Edit    {
    email                 string($email)
                          title: Email address
                          maxLength: 254
    first_name            string
                          title: First name
                          maxLength: 150
    last_name             string
                          title: Last name
                          maxLength: 150
    phone_number*         string
                          title: Phone number
                          maxLength: 12
                          minLength: 1
    avatar                string($uri)
                          title: Avatar
                          readOnly: true

}
```

POST `/accounts/login/` `accounts_login_create`

Takes a set of user credentials and returns an access and refresh JSON web token pair to prove the authentication of those credentials.

Parameters

Try it out

Name	Description
data * required `object` *(body)*	Example Value **Model**

```
TokenObtainPair    {
    username*             string
                          title: Username
                          minLength: 1
    password*             string
                          title: Password
                          minLength: 1

}
```

Responses

Response content type application/json

Code	Description
201	Example Value **Model**

```
TokenObtainPair  {
    username*           string
                        title: Username
                        minLength: 1

    password*           string
                        title: Password
                        minLength: 1

}
```

POST `/accounts/signup/` `accounts_signup_create`

Registers user in the database with given valid fields.

Parameters

Try it out

Name	Description
data * required object *(body)*	Example Value **Model**

```
SignUp  {
    username*           string
                        title: Username
                        pattern: ^[\w.@+-]+$
                        maxLength: 150
                        minLength: 1

                        Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

    password*           string
                        title: Password
                        minLength: 1
    password2*          string
                        title: Password2
                        minLength: 1
    email*              string($email)
                        title: Email
                        minLength: 1
    first_name*         string
                        title: First name
                        maxLength: 150
    last_name*          string
                        title: Last name
                        maxLength: 150
    phone_number*       string
                        title: Phone number
                        maxLength: 12
                        minLength: 1

}
```

Responses

Response content type application/json

Code	Description
201	Example Value **Model**

```
SignUp   {
    username*              string
                           title: Username
                           pattern: ^[\w.@+-]+$
                           maxLength: 150
                           minLength: 1

                           Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

    password*              string
                           title: Password
                           minLength: 1
    password2*             string
                           title: Password2
                           minLength: 1
    email*                 string($email)
                           title: Email
                           minLength: 1
    first_name*            string
                           title: First name
                           maxLength: 150
    last_name*             string
                           title: Last name
                           maxLength: 150
    phone_number*          string
                           title: Phone number
                           maxLength: 12
                           minLength: 1
    avatar                 string($uri)
                           title: Avatar
                           readOnly: true

}
```

blogs

GET /blogs/ **blogs_list**

All users can retrieve list of all available blogs.

Parameters

Try it out

Name	Description
page integer *(query)*	A page number within the paginated result set. page - A page number within the paginated re

Responses

Response content type application/json

Code	Description
200	Example Value **Model**

```
    {
    count*                 integer
    next                   string($uri)
                           x-nullable: true
    previous               string($uri)
                           x-nullable: true
    results*
                              [BlogList    {
                              id                     integer
                                                     title: ID
                                                     readOnly: true
                              restaurant*            integer
                                                     title: Restaurant
                              title*                 string
                                                     title: Title
                                                     maxLength: 100
                                                     minLength: 1
                              content*               string
                                                     title: Content
                                                     minLength: 1
                              date_created           string($date)
                                                     title: Date created
                                                     readOnly: true
                              image                  string($uri)
                                                     title: Image
                                                     readOnly: true
                                                     x-nullable: true
                              likes*                 integer
                                                     title: Likes

                           }]

 }
```

POST /blogs/add/ **blogs_add_create**

An authenicated owner can post a blog to their restaurant by specifying the fields.

Parameters

Try it out

Name	Description
data * required object *(body)*	Example Value **Model**

```
Blog    {
    title*                 string
                           title: Title
                           maxLength: 100
                           minLength: 1
    content*               string
                           title: Content
                           minLength: 1

}
```

Responses

Response content type application/json

Code	Description
201	Example Value **Model**

```
Blog   {
    restaurant            string
                          title: Restaurant
                          readOnly: true
    title*                string
                          title: Title
                          maxLength: 100
                          minLength: 1
    content*              string
                          title: Content
                          minLength: 1
    image                 string($uri)
                          title: Image
                          readOnly: true
                          x-nullable: true

}
```

GET `/blogs/feed/` `blogs_feed_list`

An authenticated user can retrieve list of all the blogs from restaurants that they follow.

Parameters

Try it out

Name	Description
page `integer` *(query)*	A page number within the paginated result set. page - A page number within the paginated re

Responses

Response content type application/json

Code	Description

Code	Description
200	Example Value **Model**

```
{
count*                 integer
next                   string($uri)
                       x-nullable: true
previous               string($uri)
                       x-nullable: true
results*
                          [BlogList   {
                          id                    integer
                                                title: ID
                                                readOnly: true
                          restaurant*           integer
                                                title: Restaurant
                          title*                string
                                                title: Title
                                                maxLength: 100
                                                minLength: 1
                          content*              string
                                                title: Content
                                                minLength: 1
                          date_created          string($date)
                                                title: Date created
                                                readOnly: true
                          image                 string($uri)
                                                title: Image
                                                readOnly: true
                                                x-nullable: true
                          likes*                integer
                                                title: Likes

                        }]

}
```

GET `/blogs/{id}/` `blogs_read`

An authenticated user can retrieve info about a blog by specifying the blog id in the url

Parameters

Try it out

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this blog. id - A unique integer value identifying this blo

Responses

Response content type application/json

Code	Description

Code	Description
200	Example Value **Model**

```
BlogDetail  {
    id                  integer
                        title: ID
                        readOnly: true
    restaurant*         integer
                        title: Restaurant
    title*              string
                        title: Title
                        maxLength: 100
                        minLength: 1
    content*            string
                        title: Content
                        minLength: 1
    date_created        string($date)
                        title: Date created
                        readOnly: true
    image               string($uri)
                        title: Image
                        readOnly: true
                        x-nullable: true
    likes*              integer
                        title: Likes

}
```

PUT `/blogs/{id}/` **blogs_update**

An authenticated owner can update it's blog by specifying id in the url and all required fields

Parameters Try it out

Name | **Description**

data * required
`object`
(body)

Example Value **Model**

```
BlogDetail  {
    restaurant*         integer
                        title: Restaurant
    title*              string
                        title: Title
                        maxLength: 100
                        minLength: 1
    content*            string
                        title: Content
                        minLength: 1
    likes*              integer
                        title: Likes

}
```

id * required
`integer`
(path)

A unique integer value identifying this blog.

id - A unique integer value identifying this blo

Responses

Response content type application/json

Code **Description**

200

Example Value **Model**

```
BlogDetail  {
    id                      integer
                            title: ID
                            readOnly: true
    restaurant*             integer
                            title: Restaurant
    title*                  string
                            title: Title
                            maxLength: 100
                            minLength: 1
    content*                string
                            title: Content
                            minLength: 1
    date_created            string($date)
                            title: Date created
                            readOnly: true
    image                   string($uri)
                            title: Image
                            readOnly: true
                            x-nullable: true
    likes*                  integer
                            title: Likes

}
```

PATCH `/blogs/{id}/` `blogs_partial_update`

An authenticated owner can partially update it's blog by specifying id in the url and all desired fields

Parameters

Try it out

Name **Description**

data * required
`object`
(body)

Example Value **Model**

```
BlogDetail  {
    restaurant*             integer
                            title: Restaurant
    title*                  string
                            title: Title
                            maxLength: 100
                            minLength: 1
    content*                string
                            title: Content
                            minLength: 1
    likes*                  integer
                            title: Likes

}
```

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this blog. id - A unique integer value identifying this blo

Responses Response content type **application/json**

Code	Description
200	Example Value **Model**

BlogDetail {

```
    id                  integer
                        title: ID
                        readOnly: true
    restaurant*         integer
                        title: Restaurant
    title*              string
                        title: Title
                        maxLength: 100
                        minLength: 1
    content*            string
                        title: Content
                        minLength: 1
    date_created        string($date)
                        title: Date created
                        readOnly: true
    image               string($uri)
                        title: Image
                        readOnly: true
                        x-nullable: true
    likes*              integer
                        title: Likes

}
```

DELETE `/blogs/{id}/` `blogs_delete`

An authenticated owner can delete their blog by specifying the blog id.

Parameters Try it out

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this blog. id - A unique integer value identifying this blo

Responses

Response content type application/json

Code	Description
204	

POST /blogs/{id}/like/ blogs_like_create

An authenticated user can like a blog post if they have not already done so.

Parameters

Try it out

Name	Description

data * required
object
(body)
	Example Value **Model**
	BlogLike {
	}

id * required integer *(path)*	A unique integer value identifying this likes blog. id - A unique integer value identifying this like

Responses

Response content type application/json

Code	Description

201	Example Value **Model**
	BlogLike {
	}

DELETE /blogs/{id}/unlike/ blogs_unlike_delete

An authenticated user can unlike a blog post they have currently liked.

Parameters Try it out

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this likes blog. id - A unique integer value identifying this like

Responses Response content type application/json

Code	Description
204	

notifications

POST `/notifications/add/` `notifications_add_create`

Creates a new notification for the specified user with the specified description

Parameters Try it out

Name	Description
data * required `object` *(body)*	Example Value **Model**

```
AddNotification  {
    user              integer
                      title: User
                      x-nullable: true
    description*      string
                      title: Description
                      maxLength: 200
                      minLength: 1

}
```

Responses Response content type application/json

Code	Description

Code	Description
201	Example Value **Model**

```
AddNotification  {
    user                integer
                        title: User
                        x-nullable: true
    description*        string
                        title: Description
                        maxLength: 200
                        minLength: 1

}
```

GET `/notifications/view/` `notifications_view_list`

Retrieves list of notification of authenticated user

Parameters

Try it out

Name	Description
page `integer` *(query)*	A page number within the paginated result set. page - A page number within the paginated re

Responses

Response content type application/json

Code	Description

Code	Description
200	Example Value **Model**

```
{
count*                  integer
next                    string($uri)
                        x-nullable: true
previous                string($uri)
                        x-nullable: true
results*
                          [Notification    {
                          id                        string
                                                    title: Id
                                                    readOnly: true
                          receiver                  string
                                                    title: Receiver
                                                    readOnly: true
                                                    minLength: 1
                          description*              string
                                                    title: Description
                                                    maxLength: 200
                                                    minLength: 1

                        }]

}
```

restaurants

POST `/restaurants/create/` restaurants_create_create

An authenticated user can create at most one restaurant, of which I will become the owner.
A restaurant is created by specifying its name, address, logo, postal code, and phone number.

Parameters Try it out

Name	Description

Name	Description
data * required object *(body)*	Example Value **Model**

```
Restaurant  {
   name*                string
                        title: Name
                        maxLength: 200
                        minLength: 1
   phone_number*        string
                        title: Phone number
                        maxLength: 12
                        minLength: 1
   address*             string
                        title: Address
                        maxLength: 200
                        minLength: 1
   postal_code*         string
                        title: Postal code
                        maxLength: 200
                        minLength: 1
   description          string
                        title: Description
                        maxLength: 200
                        x-nullable: true
   schedule             string
                        title: Schedule
                        maxLength: 200
                        x-nullable: true
   rating               integer
                        title: Rating
                        x-nullable: true

}
```

Responses

Response content type application/json

Code	Description

Code	Description
201	Example Value **Model**

```
Restaurant    {
    id                   string
                         title: Id
                         readOnly: true
    owner                string
                         title: Owner
                         readOnly: true
                         minLength: 1
    name*                string
                         title: Name
                         maxLength: 200
                         minLength: 1
    phone_number*        string
                         title: Phone number
                         maxLength: 12
                         minLength: 1
    address*             string
                         title: Address
                         maxLength: 200
                         minLength: 1
    postal_code*         string
                         title: Postal code
                         maxLength: 200
                         minLength: 1
    description          string
                         title: Description
                         maxLength: 200
                         x-nullable: true
    schedule             string
                         title: Schedule
                         maxLength: 200
                         x-nullable: true
    logo                 string($uri)
                         title: Logo
                         readOnly: true
    rating               integer
                         title: Rating
                         x-nullable: true

}
```

PUT `/restaurants/edit/{id}/` `restaurants_edit_update`

An authenticeted owner can edit the general information of their restaurant.

Parameters

Try it out

Name	Description

Name **Description**

data * required
`object`
(body)

Example Value **Model**

```
Restaurant    {
    name*               string
                        title: Name
                        maxLength: 200
                        minLength: 1
    phone_number*       string
                        title: Phone number
                        maxLength: 12
                        minLength: 1
    address*            string
                        title: Address
                        maxLength: 200
                        minLength: 1
    postal_code*        string
                        title: Postal code
                        maxLength: 200
                        minLength: 1
    description         string
                        title: Description
                        maxLength: 200
                        x-nullable: true
    schedule            string
                        title: Schedule
                        maxLength: 200
                        x-nullable: true
    rating              integer
                        title: Rating
                        x-nullable: true

}
```

id * required
`integer`
(path)

A unique integer value identifying this restaurant.

id - A unique integer value identifying this res

Responses **Response content type** application/json

Code **Description**

Code	Description
200	Example Value **Model**

```
Restaurant    {
    id                    string
                          title: Id
                          readOnly: true
    owner                 string
                          title: Owner
                          readOnly: true
                          minLength: 1
    name*                 string
                          title: Name
                          maxLength: 200
                          minLength: 1
    phone_number*         string
                          title: Phone number
                          maxLength: 12
                          minLength: 1
    address*              string
                          title: Address
                          maxLength: 200
                          minLength: 1
    postal_code*          string
                          title: Postal code
                          maxLength: 200
                          minLength: 1
    description           string
                          title: Description
                          maxLength: 200
                          x-nullable: true
    schedule              string
                          title: Schedule
                          maxLength: 200
                          x-nullable: true
    logo                  string($uri)
                          title: Logo
                          readOnly: true
    rating                integer
                          title: Rating
                          x-nullable: true

}
```

PATCH `/restaurants/edit/{id}/` `restaurants_edit_partial_update`

An authenticeted owner can edit the general information of their restaurant by specifiing the fields.

Parameters

Try it out

Name	Description

Name	Description
data * required object *(body)*	Example Value **Model**

```
Restaurant   {
    name*                  string
                           title: Name
                           maxLength: 200
                           minLength: 1
    phone_number*          string
                           title: Phone number
                           maxLength: 12
                           minLength: 1
    address*               string
                           title: Address
                           maxLength: 200
                           minLength: 1
    postal_code*           string
                           title: Postal code
                           maxLength: 200
                           minLength: 1
    description            string
                           title: Description
                           maxLength: 200
                           x-nullable: true
    schedule               string
                           title: Schedule
                           maxLength: 200
                           x-nullable: true
    rating                 integer
                           title: Rating
                           x-nullable: true

}
```

Name	Description
id * required integer *(path)*	A unique integer value identifying this restaurant. id - A unique integer value identifying this res

Responses Response content type application/json

Code	Description

Code	Description
200	Example Value **Model**

```
Restaurant  {
   id                  string
                       title: Id
                       readOnly: true
   owner               string
                       title: Owner
                       readOnly: true
                       minLength: 1
   name*               string
                       title: Name
                       maxLength: 200
                       minLength: 1
   phone_number*       string
                       title: Phone number
                       maxLength: 12
                       minLength: 1
   address*            string
                       title: Address
                       maxLength: 200
                       minLength: 1
   postal_code*        string
                       title: Postal code
                       maxLength: 200
                       minLength: 1
   description         string
                       title: Description
                       maxLength: 200
                       x-nullable: true
   schedule            string
                       title: Schedule
                       maxLength: 200
                       x-nullable: true
   logo                string($uri)
                       title: Logo
                       readOnly: true
   rating              integer
                       title: Rating
                       x-nullable: true

}
```

POST `/restaurants/images/add/` `restaurants_images_add_create` 🔓

An authenticated owner can add images to their restaurant.

Parameters Try it out

Name	Description

data * required
`object`
(body)
	Example Value **Model**
	
	Image {
	
	}

Responses Response content type application/json

Code	Description
201	Example Value **Model**

```
Image   {
    restaurant          string
                        title: Restaurant
                        readOnly: true
                        minLength: 1
    image               string($uri)
                        title: Image
                        readOnly: true

}
```

DELETE `/restaurants/images/remove/{id}/` `restaurants_images_remove_delete`

An authenticated owner can delete an image of their restaurant by specifying image id in the url.

Parameters Try it out

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this image. id - A unique integer value identifying this ima

Responses Response content type application/json

Code	Description
204	

GET `/restaurants/info/{id}/` `restaurants_info_read`

An authenticated user can retrieve info about a restaurant by specifying restaurant id in the url

Parameters Try it out

Name	Description
id * required integer *(path)*	A unique integer value identifying this restaurant. id - A unique integer value identifying this res

Responses

Response content type application/json

Code	Description
200	Example Value **Model**

```
RestaurantInfo  {
    id                  string
                        title: Id
                        readOnly: true
    owner               string
                        title: Owner
                        readOnly: true
                        minLength: 1
    name*               string
                        title: Name
                        maxLength: 200
                        minLength: 1
    phone_number*       string
                        title: Phone number
                        maxLength: 12
                        minLength: 1
    address*            string
                        title: Address
                        maxLength: 200
                        minLength: 1
    postal_code*        string
                        title: Postal code
                        maxLength: 200
                        minLength: 1
    description         string
                        title: Description
                        maxLength: 200
                        x-nullable: true
    schedule            string
                        title: Schedule
                        maxLength: 200
                        x-nullable: true
    logo                string($uri)
                        title: Logo
                        readOnly: true
    rating              integer
                        title: Rating
                        x-nullable: true
    followers*          integer
                        title: Followers
    likes*              integer
                        title: Likes
    images*                 [Image  {
                            restaurant          string
                                                title: Restaurant
                                                readOnly: true
                                                minLength: 1
                            image               string($uri)
                                                title: Image
                                                readOnly: true

                          }]
    menu*                   [Menu  {
                            restaurant          string
```

Code

```
                                   title: Restaurant
                                   readOnly: true
                                   minLength: 1
                image              string($uri)
                                   title: Image
                                   readOnly: true
                name*              string
                                   title: Name
                                   maxLength: 200
                                   minLength: 1
                description*       string
                                   title: Description
                                   maxLength: 200
                                   minLength: 1
                price*             string($decimal)
                                   title: Price

              }]
    blogs*      [Blog   {
                restaurant         string
                                   title: Restaurant
                                   readOnly: true
                title*             string
                                   title: Title
                                   maxLength: 100
                                   minLength: 1
                content*           string
                                   title: Content
                                   minLength: 1
                image              string($uri)
                                   title: Image
                                   readOnly: true
                                   x-nullable: true

              }]
    comments*   [Comments   {
                user_id            string
                                   title: User id
                                   readOnly: true
                description*       string
                                   title: Description
                                   maxLength: 200
                                   minLength: 1

              }]

 }
```

POST `/restaurants/menu/add/` `restaurants_menu_add_create`

An authenticated owner can add a menu item to their menu.

Parameters

Try it out

Name	Description

Name	Description
data * required object *(body)*	Example Value **Model**

```
Menu    {
    name*                      string
                               title: Name
                               maxLength: 200
                               minLength: 1
    description*               string
                               title: Description
                               maxLength: 200
                               minLength: 1
    price*                     string($decimal)
                               title: Price

}
```

Responses Response content type application/json

Code	Description
201	Example Value **Model**

```
Menu    {
    restaurant                 string
                               title: Restaurant
                               readOnly: true
                               minLength: 1
    image                      string($uri)
                               title: Image
                               readOnly: true
    name*                      string
                               title: Name
                               maxLength: 200
                               minLength: 1
    description*               string
                               title: Description
                               maxLength: 200
                               minLength: 1
    price*                     string($decimal)
                               title: Price

}
```

PUT `/restaurants/menu/update/{id}/` `restaurants_menu_update_update`

An authenticated owner can update a menu item on their menu.

Parameters Try it out

Name	Description

Name	Description
data * required object *(body)*	Example Value **Model**

```
Menu    {
    name*                   string
                            title: Name
                            maxLength: 200
                            minLength: 1
    description*            string
                            title: Description
                            maxLength: 200
                            minLength: 1
    price*                  string($decimal)
                            title: Price

}
```

Name	Description
id * required integer *(path)*	A unique integer value identifying this menu. id - A unique integer value identifying this me

Responses Response content type application/json

Code	Description
200	Example Value **Model**

```
Menu    {
    restaurant              string
                            title: Restaurant
                            readOnly: true
                            minLength: 1
    image                   string($uri)
                            title: Image
                            readOnly: true
    name*                   string
                            title: Name
                            maxLength: 200
                            minLength: 1
    description*            string
                            title: Description
                            maxLength: 200
                            minLength: 1
    price*                  string($decimal)
                            title: Price

}
```

PATCH /restaurants/menu/update/{id}/ restaurants_menu_update_partial_update

An authenticated owner can update a menu item on their menu by specifying fields.

Parameters

Try it out

Name	Description
data * required object *(body)*	Example Value **Model**

```
Menu  {
    name*                  string
                           title: Name
                           maxLength: 200
                           minLength: 1
    description*           string
                           title: Description
                           maxLength: 200
                           minLength: 1
    price*                 string($decimal)
                           title: Price

}
```

Name	Description
id * required integer *(path)*	A unique integer value identifying this menu. id - A unique integer value identifying this me

Responses

Response content type application/json

Code	Description
200	Example Value **Model**

```
Menu  {
    restaurant             string
                           title: Restaurant
                           readOnly: true
                           minLength: 1
    image                  string($uri)
                           title: Image
                           readOnly: true
    name*                  string
                           title: Name
                           maxLength: 200
                           minLength: 1
    description*           string
                           title: Description
                           maxLength: 200
                           minLength: 1
    price*                 string($decimal)
                           title: Price

}
```

GET /restaurants/search/ restaurants_search_list

Authenticated user can search through restaurants by their name, foods, or address. Most popular restaurants should come up first in the results.

You could search by sending a query parameter in the url with the key 'search'.

Parameters Try it out

Name	Description
search string *(query)*	A search term. search - A search term.
page integer *(query)*	A page number within the paginated result set. page - A page number within the paginated re

Responses Response content type application/json

Code	Description
200	Example Value **Model**

```
{
count*                      integer
next                        string($uri)
                            x-nullable: true
previous                    string($uri)
                            x-nullable: true
results*                        [SearchRestaurant   {
                                id                      integer
                                                        title: ID
                                                        readOnly: true
                                name*                   string
                                                        title: Name
                                                        maxLength: 200
                                                        minLength: 1
                                address*                string
                                                        title: Address
                                                        maxLength: 200
                                                        minLength: 1
                                followers*              integer
                                                        title: Followers

                              }]

}
```

POST /restaurants/{id}/comment/ restaurants_comment_create

An authenticated user can comment on a restaurant by specifying restaurant id in the url and description field in the body.

Parameters

Try it out

Name	Description

data * required
`object`
(body)
	Example Value **Model**
	CommentRestaurant {
	`description*` string
	title: Description
	maxLength: 200
	minLength: 1
	}

id * required `integer` *(path)*	A unique integer value identifying this comments. id - A unique integer value identifying this cor

Responses

Response content type application/json

Code	Description

201	Example Value **Model**
	CommentRestaurant {
	`description*` string
	title: Description
	maxLength: 200
	minLength: 1
	}

POST `/restaurants/{id}/follow/` `restaurants_follow_create`

Authenticated user can follow another restaurant by specifying restaurant id in the url.

Parameters

Try it out

Name	Description

Name	Description

data * required
`object`
(body)
	Example Value **Model**
	RestaurantFollow {
	user integer
	title: User
	x-nullable: true
	}

id * required `integer` *(path)*	A unique integer value identifying this likes restaurant. id - A unique integer value identifying this like

Responses

Response content type application/json

Code	Description

201	Example Value **Model**
	RestaurantFollow {
	user integer
	title: User
	x-nullable: true
	}

POST `/restaurants/{id}/like/` `restaurants_like_create`

Authenticated user can like another restaurant by specifying restaurant id in the url.

Parameters

Try it out

Name	Description

data * required
`object`
(body)
	Example Value **Model**
	RestaurantLike {
	user integer
	title: User
	x-nullable: true
	}

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this likes restaurant. id - A unique integer value identifying this like

Responses Response content type application/json

Code	Description
201	Example Value **Model**

```
RestaurantLike    {
    user              integer
                      title: User
                      x-nullable: true

}
```

DELETE `/restaurants/{id}/unfollow/` `restaurants_unfollow_delete`

Authenticated user can unfollow another restaurant by specifying id in the url.

Parameters Try it out

Name	Description
id * required `integer` *(path)*	A unique integer value identifying this follows. id - A unique integer value identifying this foll

Responses Response content type application/json

Code	Description
204	

DELETE `/restaurants/{id}/unlike/` `restaurants_unlike_delete`

Authenticated user can unlike another restaurant by specifying id in the url.

Parameters

Try it out

Name	Description
id * required integer *(path)*	A unique integer value identifying this likes restaurant. id - A unique integer value identifying this like

Responses

Response content type application/json

Code	Description
204	

Models

```
Edit    {
   email                string($email)
                        title: Email address
                        maxLength: 254
   first_name           string
                        title: First name
                        maxLength: 150
   last_name            string
                        title: Last name
                        maxLength: 150
   phone_number*        string
                        title: Phone number
                        maxLength: 12
                        minLength: 1
   avatar               string($uri)
                        title: Avatar
                        readOnly: true

}
```

```
TokenObtainPair    {
   username*            string
                        title: Username
                        minLength: 1
   password*            string
                        title: Password
                        minLength: 1

}
```

SignUp {

```
    username*                string
                             title: Username
                             pattern: ^[\w.@+-]+$
                             maxLength: 150
                             minLength: 1

                             Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

    password*                string
                             title: Password
                             minLength: 1
    password2*               string
                             title: Password2
                             minLength: 1
    email*                   string($email)
                             title: Email
                             minLength: 1
    first_name*              string
                             title: First name
                             maxLength: 150
    last_name*               string
                             title: Last name
                             maxLength: 150
    phone_number*            string
                             title: Phone number
                             maxLength: 12
                             minLength: 1
    avatar                   string($uri)
                             title: Avatar
                             readOnly: true

}
```

BlogList {

```
    id                       integer
                             title: ID
                             readOnly: true
    restaurant*              integer
                             title: Restaurant
    title*                   string
                             title: Title
                             maxLength: 100
                             minLength: 1
    content*                 string
                             title: Content
                             minLength: 1
    date_created             string($date)
                             title: Date created
                             readOnly: true
    image                    string($uri)
                             title: Image
                             readOnly: true
                             x-nullable: true
    likes*                   integer
                             title: Likes

}
```

```
Blog    {
    restaurant              string
                            title: Restaurant
                            readOnly: true
    title*                  string
                            title: Title
                            maxLength: 100
                            minLength: 1
    content*                string
                            title: Content
                            minLength: 1
    image                   string($uri)
                            title: Image
                            readOnly: true
                            x-nullable: true

}
```

```
BlogDetail    {
    id                      integer
                            title: ID
                            readOnly: true
    restaurant*             integer
                            title: Restaurant
    title*                  string
                            title: Title
                            maxLength: 100
                            minLength: 1
    content*                string
                            title: Content
                            minLength: 1
    date_created            string($date)
                            title: Date created
                            readOnly: true
    image                   string($uri)
                            title: Image
                            readOnly: true
                            x-nullable: true
    likes*                  integer
                            title: Likes

}
```

```
BlogLike    {

}
```

```
AddNotification    {
    user                    integer
                            title: User
                            x-nullable: true
    description*            string
                            title: Description
                            maxLength: 200
                            minLength: 1

}
```

```
Notification  {
    id                  string
                        title: Id
                        readOnly: true
    receiver            string
                        title: Receiver
                        readOnly: true
                        minLength: 1
    description*        string
                        title: Description
                        maxLength: 200
                        minLength: 1

}
```

```
Restaurant  {
    id                  string
                        title: Id
                        readOnly: true
    owner               string
                        title: Owner
                        readOnly: true
                        minLength: 1
    name*               string
                        title: Name
                        maxLength: 200
                        minLength: 1
    phone_number*       string
                        title: Phone number
                        maxLength: 12
                        minLength: 1
    address*            string
                        title: Address
                        maxLength: 200
                        minLength: 1
    postal_code*        string
                        title: Postal code
                        maxLength: 200
                        minLength: 1
    description         string
                        title: Description
                        maxLength: 200
                        x-nullable: true
    schedule            string
                        title: Schedule
                        maxLength: 200
                        x-nullable: true
    logo                string($uri)
                        title: Logo
                        readOnly: true
    rating              integer
                        title: Rating
                        x-nullable: true

}
```

```
Image  {
    restaurant          string
                        title: Restaurant
                        readOnly: true
                        minLength: 1
    image               string($uri)
                        title: Image
                        readOnly: true

}
```

```
Menu    {
   restaurant              string
                           title: Restaurant
                           readOnly: true
                           minLength: 1
   image                   string($uri)
                           title: Image
                           readOnly: true
   name*                   string
                           title: Name
                           maxLength: 200
                           minLength: 1
   description*            string
                           title: Description
                           maxLength: 200
                           minLength: 1
   price*                  string($decimal)
                           title: Price

}
```

```
Comments    {
   user_id                 string
                           title: User id
                           readOnly: true
   description*            string
                           title: Description
                           maxLength: 200
                           minLength: 1

}
```

```
RestaurantInfo    {
   id                      string
                           title: Id
                           readOnly: true
   owner                   string
                           title: Owner
                           readOnly: true
                           minLength: 1
   name*                   string
                           title: Name
                           maxLength: 200
                           minLength: 1
   phone_number*           string
                           title: Phone number
                           maxLength: 12
                           minLength: 1
   address*                string
                           title: Address
                           maxLength: 200
                           minLength: 1
   postal_code*            string
                           title: Postal code
                           maxLength: 200
                           minLength: 1
   description             string
                           title: Description
                           maxLength: 200
                           x-nullable: true
   schedule                string
                           title: Schedule
                           maxLength: 200
                           x-nullable: true
   logo                    string($uri)
                           title: Logo
                           readOnly: true
   rating                  integer
                           title: Rating
                           x-nullable: true
   followers*              integer
                           title: Followers
   likes*                  integer
```

```
                             title: Likes
      images*
                               [Image    {
                               restaurant             string
                                                      title: Restaurant
                                                      readOnly: true
                                                      minLength: 1
                               image                  string($uri)
                                                      title: Image
                                                      readOnly: true

                             }]
      menu*
                               [Menu    {
                               restaurant             string
                                                      title: Restaurant
                                                      readOnly: true
                                                      minLength: 1
                               image                  string($uri)
                                                      title: Image
                                                      readOnly: true
                               name*                  string
                                                      title: Name
                                                      maxLength: 200
                                                      minLength: 1
                               description*           string
                                                      title: Description
                                                      maxLength: 200
                                                      minLength: 1
                               price*                 string($decimal)
                                                      title: Price

                             }]
      blogs*
                               [Blog    {
                               restaurant             string
                                                      title: Restaurant
                                                      readOnly: true
                               title*                 string
                                                      title: Title
                                                      maxLength: 100
                                                      minLength: 1
                               content*               string
                                                      title: Content
                                                      minLength: 1
                               image                  string($uri)
                                                      title: Image
                                                      readOnly: true
                                                      x-nullable: true

                             }]
      comments*
                               [Comments    {
                               user_id                string
                                                      title: User id
                                                      readOnly: true
                               description*           string
                                                      title: Description
                                                      maxLength: 200
                                                      minLength: 1

                             }]

  }
```

```
SearchRestaurant    {
    id                  integer
                        title: ID
                        readOnly: true
    name*               string
                        title: Name
                        maxLength: 200
                        minLength: 1
    address*            string
                        title: Address
                        maxLength: 200
                        minLength: 1
    followers*          integer
                        title: Followers

}
```

```
CommentRestaurant    {
    description*        string
                        title: Description
                        maxLength: 200
                        minLength: 1

}
```

```
RestaurantFollow    {
    user                integer
                        title: User
                        x-nullable: true

}
```

```
RestaurantLike    {
    user                integer
                        title: User
                        x-nullable: true

}
```