Restify API

A collection of API calls for the Restify API. Restify is an app designated for hosting and reserving properties.

Note for TA: When using the shell scripts, please add yourself to the sudoers list and use source in front of the script. For example, source setup.sh.

Reservations

API calls related to creating, viewing, and changing the status of reservations.

GET Guest Reservation List

```
http://localhost:8000/reservations/guest/reservations/?status=Canceled&page=1
```

Returns the reservation list for a user from the guest point of view. Essentially the reservations made where the current user is a guest.

AUTHORIZATION Bearer Token

Token	<token>

PARAMS

status	Canceled The status filter you would like to apply
page	1 Page Number

GET Host Reservation List

```
http://localhost:8000/reservations/host/reservations/?page=1&status=Pending
```

Returns the reservation list from the host point of view. Essentially the list of reservations on your properties as a host.

AUTHORIZATION Bearer Token

Token	<token>

PARAMS

page	1 Page number
status	Pending The status you want to filter reservations by

POST Create Reservation

```
http://localhost:8000/reservations/:property-id/reserve/
```

Creates a reservation for the property with property-id with the given date range.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

property-id	1 The id of the property you are trying to reserve for

Body formdata

start_date	2023-03-17 The start date of your reservation
end_date	2023-03-18 The end date of your reservation

GET Cancel Reservation/Request Cancel

```
http://localhost:8000/reservations/:reservation-id/cancel/
```

Allows hosts and guests to cancel their reservations. If pending, then guests can cancel at any time. If already approved, then guests must send a request to cancel which the host can approve using this call. The host can cancel a reservation at any time.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

reservation-id	2 The id for a reservation you want to cancel

GET Approve Reservation

```
http://localhost:8000/reservations/:reservation-id/approve/
```

Allows hosts to approve reservation requests from guests. The status of these reservations changes accordingly.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

reservation-id	2 The id for a reservation you want to approve

GET Deny Reservation

```
http://localhost:8000/reservations/:reservation-id/deny/
```

Allows hosts to deny reservation requests. The status changes from pending to denied.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

reservation-id	2 The id for a reservation you want to deny

GET Terminate Reservation

```
http://localhost:8000/reservations/:reservation-id/terminate/
```

Allows hosts to terminate reservations. Status will change to terminated.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

reservation-id	2 The id for a reservation you want to terminate

Tokens

API calls related to tokens.

POST Refresh token

```
http://localhost:8000/api/token/refresh/
```

Allows users to refresh their tokens.

Body formdata

refresh	The refresh token for this user.

Comments

Comments

API calls related to creating and viewing comments on properties and users.

GET Property Comments

```
http://localhost:8000/social/comments/property/:property-id/?page=1
```

Returns paginated property comments.

PARAMS

page	1 Page number

PATH VARIABLES

property-id	2 The id of the property whose comments you want to read

GET User Comments

```
http://localhost:8000/social/comments/user/:user-id/?page=1
```

Returns paginated user comments.

PARAMS

page	1 Page number

PATH VARIABLES

user-id	2 The id of the user you want to receive comments about

POST Guest Comment on Property

```
http://localhost:8000/social/comments/property/write/:reservation-id/
```

Allows a guest to comment on a specific property that they stayed at with reservation-id.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

reservation-id	9 The id of the reservation the guest is trying to comment on

Body formdata

message	great house and host The message in the comment.

POST Reply to Comment on Property

```
http://localhost:8000/social/comments/property/reply/:comment-id/
```

Allows the host to reply to a comment by a user on their property, or the user to reply to reply left by the host on their original comment.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

comment-id	6 The property comment you are replying to

Body formdata

message	No problem The message in the comment.

POST Host Comment on User

```
http://localhost:8000/social/comments/user/write/:reservation-id/
```

Allows the host to comment on a user who stayed at their property with reservation-id.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

reservation-id	5 The id of the reservation where you hosted this guest

Body formdata

message	Bad guest The message in the comment.

POST Reply to Comment on User

```
http://localhost:8000/social/comments/user/reply/:comment-id/
```

Allows users to respond to comments on their profile.

AUTHORIZATION Bearer Token

Token	<token>

PATH VARIABLES

comment-id	9 The host's comment you are replying to

Body formdata

message	Thanks The message in the comment.

Properties

POST Get Access Token

```
http://127.0.0.1:8000/api/token/
```

Gets access token for a user with the given username and password.

Body formdata

username	Joe11 Username of pre-existing user
password	123 Password of pre-existing user

POST Create Property

```
http://127.0.0.1:8000/properties/createproperty/
```

Creates a property with given parameters. Request user must be logged in to create a property. Host value should match primary key of request user, otherwise 403 Unauthorized is returned.

AUTHORIZATION Bearer Token

Token

Body formdata

host	1 Primary key value of host user
name	Maryum's House Name of property
address	Mississauga, Canada Address of property
preview	Preview image of property

PUT Update Property

```
http://127.0.0.1:8000/properties/:property_id/updateproperty/
```

Updates attributes of a property if request user is host of property.

AUTHORIZATION Bearer Token

Token

PATH VARIABLES

property_id	Primary key of property to be updated

Body formdata

host	1 Primary key value of host user
name	Updated House Updated name of property
address	Milton, Canada Updated address of property
preview	Preview image of property. Must include image file even when the preview stays the same.

GET Check Update

```
http://127.0.0.1:8000/properties/searchproperty/
```

AUTHORIZATION Bearer Token

Token

Body formdata

location	Milton Location of properties to be returned

DELETE Delete Property

```
http://127.0.0.1:8000/properties/:property_id/deleteproperty/
```

Deletes property with the given primary key if request user is host of property.

AUTHORIZATION Bearer Token

Token

PATH VARIABLES

property_id	Primary key of property to be deleted

DELETE Delete Again

```
http://127.0.0.1:8000/properties/:property_id/deleteproperty/
```

AUTHORIZATION Bearer Token

Token

PATH VARIABLES

property_id	Primary key of property to be deleted

GET Check Deleted

```
http://127.0.0.1:8000/properties/searchproperty/
```

AUTHORIZATION Bearer Token

Token

HEADERS

location	Milton

POST Add Availability

```
http://127.0.0.1:8000//properties/:property_id/addavailability/
```

Adds a date range to a property. If property does not belong to request user, returns 403 unauthorized error.

AUTHORIZATION Bearer Token

Token

PATH VARIABLES

property_id	2 Primary key value of property to add availability to

Body formdata

start_date	2023-03-28 Start date of availability. Must be in the form "YYYY-MM-DD"
end_date	2023-04-15 End date of availability. Must be in the form "YYYY-MM-DD"
price	170 Price per night within date range

GET Test Date Range

```
http://127.0.0.1:8000/properties/searchproperty/
```

Body formdata

start_date	2023-03-11
end_date	2023-03-15

GET Test Date Range TV

```
http://127.0.0.1:8000/properties/searchproperty/
```

Returns list of properties that match given query.

Body formdata

start_date	2023-03-11 Start date of availability of property
end_date	2023-03-15 End date of availability of property
tv	True Boolean value that represents whether the property should have a specific amenity. Properties can be filtered by the following amenities: tv, air_condition, wifi, kitchen, hair_dryer, heating, iron, workspace, washer, dryer, pool, free_parking, crib, grill, hot_tub, EV_charger, gym, indoor_fireplace, breakfast, smoking_allowed.

GET Test Date Range TV Order

```
http://127.0.0.1:8000/properties/searchproperty/
```

Body formdata

start_date	2023-03-11 Start date of availability of property

end_date	2023-03-15 End date of availability of property
tv	True Boolean value that represents whether the property should have a specific amenity. Properties can be filtered by the following amenities: tv, air_condition, wifi, kitchen, hair_dryer, heating, iron, workspace, washer, dryer, pool, free_parking, crib, grill, hot_tub, EV_charger, gym, indoor_fireplace, breakfast, smoking_allowed.
order_by	price Field to order the search results by. Options are: price, name or both.

POST Create Property UnAuthorized

```
http://127.0.0.1:8000/properties/createproperty/
```

Body formdata

host	1
name	Not My House
address	Toronto, Canada
guests	1

Accounts

POST Signup

```
http://localhost:8000/accounts/signup/
```

Allow new users to create an account.

Body formdata

username	user_1_username Username
first_name	User First name
last_name	One Last name
email	user.one@gmail.com Email address
password	strong_password123%23 A strong password that contains at least one uppercase letter, digit and special character
confirm_password	strong_password123%23 Retyped password that matches password
phone_number	6478982918 Phone number
photo	Profile picture of the user

POST Login

```
http://localhost:8000/api/token/
```

Allow users to be authorized by returning an access and refresh jwt token.

Body formdata

username	user_1_username Username of the user logging in
password	strong_password123%23 Password of the user logging in

PUT Update Profile

```
http://localhost:8000/accounts/editprofile/
```

Allow the current authorized user to edit their profile.

AUTHORIZATION Bearer Token

Token

Body formdata

username	user_1_username Username
first_name	John First name
last_name	One Last name
email	john.one@gmail.com Email address
password	strong_password123 A strong password that contains at least one uppercase letter, digit and special character
confirm_password	strong_password123 Retyped password that matches password
phone_number	6478982918 Phone number
photo	Profile picture

Notifications

GET User Notifications List

```
http://localhost:8000/social/notifications/?page=1
```

Return a list of notifications belonging to the current user. Takes optional argument 'read' to filter read or unread notifications.

AUTHORIZATION Bearer Token

Token <token>

PARAMS

read False

Optional filter for read or unread notifications

page 1

Page number

GET Read Notification

```
http://localhost:8000/social/notifications/:notification_id/
```

Returns details, including the message of the notification asked for.

PATH VARIABLES

notification_id 1

The id of the notification you want to read

DELETE Delete Notification

```
http://localhost:8000/social/notifications/:notification_id/delete/
```

Deletes the notification with the given notification id.

AUTHORIZATION Bearer Token

Token

PATH VARIABLES

notification_id 9

The id of the notification to be deleted

DELETE Clear Notifications

```
http://localhost:8000/social/notifications/clear/?read=True
```

Delete all notifications belonging to the current user. Optional argument 'read' to only delete read or unread notifications.

AUTHORIZATION Bearer Token

Token

PARAMS

read	True
	Optional filter for read or unread notifications