Restify

By: Group 2635

Models

The main models involved in Restify are detailed below. Child models that exist only for their parent models will be discussed in the parent model.

User

A custom `User` for authentication. `User`s are identified by a unique email, and includes a first and last name, phone number, and an avatar image.

Property

A `Property` is a location for `User`s to rent. It is owned by a `User`, and holds information, such as its location and a description. It also holds `PriceModifier`s, `PropertyImage`s, and `Amenity`s. `PriceModifier` is a `{month: x, price_modifier: y}` pair for monthly discounts, `PropertyImage` is an image of the property, and `Amenity` is a string representing a feature, such as Wi-Fi, an outdoor pool, or other such additions.

Reservation

A `Reservation` is a booking on a property for a certain time interval. It has a `User` which is the reserver, a `Property`, a start and end date, and a status, which includes keywords such as "Approved" or "Pending".

Comment

A `Comment` on a `User` or a `Property`. `Comment`s consist of a description and its `User` commenter. There are 2 subtypes of `Comment`s: `Review` and `Reply`.

`Review`s are comments that also include a rating from 1-5. `Reply`s are comments (without ratings) that are attached to a `Review`.

Hosts of `Property`s may leave `Review`s on `User`s who have stayed on their property.

Similarly, `User`s who have completed a stay at a `Property` or had their reservation cancelled may leave a `Review` on a `Property`, which can begin a `Reply` chain between the host and the guest.

Notification

`Notification`s are informational messages delivered to the user, and are composed of their string content and a `User` receiver. They are delivered on important events such as new `Reservation` requests.

API Endpoints

Accounts

SIGN UP

Endpoint:

localhost:8000/accounts/signup/

Description:

Register a new `User` account. Adding an avatar is optional.

Methods:

POST

Authentication:

No

Payload:

```
{
    "email": "",
    "first_name": "",
    "last_name": "",
    "password": "",
    "repeat_password": "",
    "phone_number": "",
    "avatar": null
}
```

LOG IN

Endpoint:

localhost:8000/api/token/

Description:

Retrieve an access token for authentication. The access token lasts for 24 hours.

Methods:

POST

Authentication:

No

Payload:

```
{
    "email": "",
    "password": ""
}
```

VIEW PROFILE

Endpoint:

localhost:8000/accounts/profile/view/<int:user_id>/

Description:

View an existing `User`'s profile. Supply the desired user's ID in the URL.

Methods:

GET

Authentication:

No

EDIT PROFILE

Endpoint:

localhost:8000/accounts/profile/edit/

Description:

Edit a `User` account.

Methods:

PUT, PATCH

Authentication:

Yes

Payload:

```
{
    "first_name": "",
    "last_name": "",
    "phone_number": "",
    "avatar": null
}
```

Comments

CREATE USER COMMENT

Endpoint:

localhost:8000/comments/user/<int:user_id>/review/create/

Description:

Leave a comment and rating on a `User` who has completed a stay at your property.

Methods:

POST

Authentication:

Yes

Payload:

```
{
    "content": "",
    "rating": null
}
```

VIEW USER COMMENTS

Endpoint:

localhost:8000/comments/user/<int:user_id>/review/

Description:

See all comments about a `User` who completed a stay at your property.

Methods:

GET

Authentication:

Yes

VIEW PROPERTY COMMENTS

Endpoint:

localhost:8000/comments/property/<int:property_id>/review/

Description:

See all comments (and their replies) on a `Property`.

Methods:

GET

Authentication:

No

CREATE NEW PROPERTY

Endpoint:

localhost:8000/properties/create/

Description:

Create a new `Property`, with its parameters set in the JSON payload.

Methods:

POST

Authentication:

Yes

Payload:

```
{
    "price_modifiers": [{"month": ..., "price_modifier": ...}, {"month": ...,
"price_modifier": ...}],
    "property_images": [...],
    "amenities": [{"name": ...}, {"name": ...}, {"name": ...}],
    "name": "...",
    "description": "...",
    "address": "...",
    "country": "...",
    "state": "...",
    "city": "...",
    "max_num_guests": ...,
    "num_beds": ...,
    "num_baths": ...,
    "nightly_price": ...,
    "owner": ...
}
```

UPDATE PROPERTY

Endpoint:

localhost:8000/properties/update/<int:pk>/

Description:

Update a `Property` with the given ID, with its parameters set in the JSON payload. Returns the `Property` with the given ID on a GET request to allow for access to its current values.

Methods:

GET, PUT, PATCH

Authentication:

Yes

Payload:

```
{
    "price_modifiers": [{"month": ..., "price_modifier": ...}, {"month": ...,
"price_modifier": ...}],
    "property_images": [...],
    "amenities": [{"name": ...}, {"name": ...}, {"name": ...}],
    "name": "...",
    "description": "...",
    "address": "...",
    "country": "...",
    "state": "...",
    "city": "...",
    "max_num_guests": ...,
    "num_beds": ...,
    "num_baths": ...,
    "nightly_price": ...,
    "owner": ...
}
```

CREATE PROPERTY COMMENT

Endpoint:

localhost:8000/comments/property/<int:property_id>/review/create/

Description:

Create a new comment thread (`Rating`) on a `Property`. Only users who have stayed at the property or had their reservations terminated by the host may leave a single comment.

Methods:

POST

Authentication:

Yes

Payload:

```
{
    "content": "",
    "rating": null
}
```

CREATE PROPERTY REPLY

Endpoint:

localhost:8000/comments/property/<int:property_id>/reply/create/

Description:

Create a new `Reply` to an existing `Rating` on a `Property`. Only the property host and the original `Rating` commenter may create a reply in alternating fashion.

Methods:

POST

Authentication:

Yes

Payload:

```
{
    "content": "",
    "reply_to": null
}
```

DELETE PROPERTY

Endpoint:

localhost:8000/properties/delete/<int:pk>/

Description:

Delete a `Property` with the given ID.

Methods:

DELETE

Authentication:

Yes

RETRIEVE PROPERTY

Endpoint:

localhost:8000/properties/retrieve/<int:pk>/

Description:

Retrieve a `Property` with the given ID.

Methods:

GET

Authentication:

No

RETRIEVE ALL PROPERTIES

Endpoint:

localhost:8000/properties/retrieve/all/

Description:

Retrieve all `Properties`.

Methods:

GET

Authentication:

No

FILTERING & ORDERING:

List of all properties can be filtered and sorted by using GET parameters. Endpoint: localhost:8000/properties/retrieve/all/?{filter/order}={param}&{filter/order}={param}

Filtering parameters: country, minPrice, maxPrice, minGuests
Ordering parameter: orderBy (choices: `[ priceASC, priceDESC, bedsASC, bedsDESC ]`)

Usage example: `localhost:8000/properties/retrieve/all/?country=Canada&minGuests=10&minPrice=550&maxPrice=675&orderBy=priceASC` gives all properties in Canada with space for at least 10 guests and nightly price between $550 and $675, sorted based on price lowest-highest

RETRIEVE ALL PROPERTIES FOR USER

Endpoint:

localhost:8000/properties/retrieve/user/

Description:

Retrieve all `Properties` owned by the current user.

Methods:

GET

Authentication:

Yes

CREATE NEW RESERVATION

Endpoint:

localhost:8000/reservations/create/

Description:

Create a new `Reservation` that is set to a `Status` of PENDING. The reservation start date, end date, property, and expiry duration are specified in the JSON payload.

Methods:

POST

Authentication:

Yes

Payload:

```
{
    "start_date": "2023-02-13",
    "end_date": "2023-02-14",
    "property": 1,
    "seconds_before_expiry": null,
}
```

RETRIEVE RESERVATIONS

Endpoint:

localhost:8000/reservations/retrieve/all/?type={...}&status={...}

Description:

Retrieve all `Reservations` with two filters: type (incoming or outgoing) and status (pending, completed, etc.).

Methods:

GET

Authentication:

Yes

APPROVE RESERVATION REQUEST

Endpoint:

localhost:8000/reservations/approve/<int:reservation_id>/

Description:

Approve a `Reservation` with the specified ID in the URL.

Methods:

PUT, PATCH

Authentication:

Yes

DENY RESERVATION REQUEST

Endpoint:

localhost:8000/reservations/deny/<int:reservation_id>/

Description:

Deny a `Reservation` with the specified ID in the URL.

Methods:

PUT, PATCH

Authentication:

Yes

Complete RESERVATION REQUEST

Endpoint:

localhost:8000/reservations/deny/<int:reservation_id>/

Description:

Set the `Reservation` with the specified ID in the URL to have status COMPLETED.

Methods:

PUT, PATCH

Authentication:

Yes

CREATE RESERVATION CANCELLATION REQUEST

Endpoint:

localhost:8000/reservations/req_cancellation/<int:reservation_id>/

Description:

Set a `Reservation` with the specified ID in the URL to have status CANCELLATION_REQUESTED.

Methods:

PUT, PATCH

Authentication:

Yes

CONFIRM RESERVATION CANCELLATION REQUEST

Endpoint:

localhost:8000/reservations/cancel/<int:reservation_id>/

Description:

Approve the `Reservation`'s cancellation request with the specified ID in the URL.

Methods:

PUT, PATCH

Authentication:

Yes

DENY RESERVATION CANCELLATION REQUEST

Endpoint:

localhost:8000/reservations/deny_cancellation_req/<int:reservation_id>

Description:

Deny the `Reservation`'s cancellation request with the specified ID in the URL.

Methods:

PUT, PATCH

Authentication:

Yes

TERMINATE RESERVATION

Endpoint:

localhost:8000/reservations/terminate/<int:reservation_id>/

Description:

Terminate the `Reservation` with the specified ID in the URL.

Methods:

PUT, PATCH

Authentication:

Yes

Notifications

VIEW LIST OF NOTIFICATIONS

Endpoint:

localhost:8000/notifications/list/

Description:

View list of notifications you have received

Methods:

GET

Authentication:

Yes

READ NOTIFICATION

Endpoint:

localhost:8000/notifications/int:notification_id

Description:

View a specific notification with id: `notification_id`

Methods:

GET

Authentication:

Yes

CLEAR NOTIFICATION

Endpoint:

localhost:8000/notifications/clear/int:notification_id/

Description:

Clear (Delete) a specific notification with id: `notification_id`

Methods:

DELETE

Authentication:

Yes

RECEIVING NOTIFICATION

Endpoint:

localhost:8000/notifications/create/

Description:

Generate a notification that can either be one of `[ host_new_reservation, host_cancellation_request, host_property_new_comment, guest_approved_reservation, guest_cancellation_request ]`

Methods:

GET, POST

Authentication:

Yes

Staff:

Yes

Payload:

```
{
    "content": "",
    "created_when": "2023-03-10",
    "receiver": ""
}
```