

RESTful API Spec

Fall 2020

Entity Details	2
Create a User	3
Update user	5
Create a listing	8
Get listings	10
Update a listing	13
Delete a listing	16
Add a listing to favorites	18
Get favorites	21
Update a favorite	23
Delete a favorite	26

DataModel and general API - Overview

Overview: The user in this API is assumed as a real-estate agent. This API allows access to the user's listings that they have created as well as any listings they have favorited. There are three tables (see below for details) that comprise this API. The user table, the listings table and the favorites tables. The relationships are as follows: The user's unique ID is attached to any listing that is created by the user. Each listing requires a user unique ID - this is done automatically. Any user can favorite any listing by using the listing unique ID. When a user favorites a listing, the user's unique ID is attached to that favorite. Every user has their own unique favorites list that they can request. The user can find the listings - along with the listing unique ID by querying all listings (see below).

Entity Details

User: Table holding the users

Name	Type	Description	Required?
fname	String	*Automatically taken from your google profile upon creation	Yes
lname	String	*Automatically taken from your google profile upon creation	Yes
agency	String	Name of current real estate agency	Yes
UID	int	*Unique user ID from google, used for relationships	Yes
id	int	*Database unique identifier	Yes

Listings: Table holding all listings

Name	Type	Description	Required?
representative	int	*Relationship: Automatically filled by user account.	Yes
price	int	Price of listing	No
sqare_feet	int	listing_properties = ['price', 'square_feet', 'bedrooms', 'bathrooms', 'keywords', 'type', 'representative']	No
bedrooms	int	Amount bedrooms in listing	No
bathrooms	int	Amount bathrooms in listing	No
keywords	List[String]	Special keywords for listing	No
Typ	String	Type of listing i.e. 'house', 'condo'	No
id	int	*Database unique identifier	Yes

Favorites: Table holding favorited listings by user

Name	Type	Description	Required?
UID	int	*Relationship: Automatically filled by the favoriting user account.	Yes
LID	int	Relationship: Unique listing database ID, get from listing object.	Yes
comments	String	Any comments about the listing i.e. "Jen will love this".	No
strength	int	Indication of the strength of favorite	No
id	int	*Database unique identifier	Yes

* indicates a field that is automatically made by either the database or the backend code. No action required by the user.

Create a User

Allows you to create a new user.

POST /users

Request

Prerequisites:

The user must first authenticate with google. The result of a successful authentication will reroute the user to '/login'. There, the user will need to attach the 'access_token' and the 'JWT' as seen below.

Path Parameters

None

Path Headers

'token' = {access_token}

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
agency	String	Name of current real estate agency	Yes

Request Body Example

```
{
  "name": "remax"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	The request is missing or adding more required attributes
Failure	401 Unauthorized	User has not been authorized through google
Failure	403 Forbidden	User already has an account
Failure	406 Accept type	Server does not support the requested media

Failure	415 unsupported media	The body payload was not labeled JSON
---------	-----------------------	---------------------------------------

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 201 Created
{
  "UID": "101062042114979755562",
  "agency": "remax",
  "fname": "Chris",
  "id": 5138716758638592,
  "lname": "Nelson",
  "self": "https://final-rest-chris.wn.r.appspot.com/user/5138716758638592"
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 403 Forbidden
{
  "Error": "Error": "You are forbidden"
}
```

Failure

```
Status: 406 bad media
{
  "Error": "Server expects application/json Content-Type"
}
```

Failure

```
Status: 415 unsupported media
{
  "Error": "Not an acceptable media type"
}
```

Update a User

Allows you to update an existing user.

PATCH or PUT /users

Request

Prerequisites:

The user must first authenticate with google. The result of a successful authentication will reroute the user to '/login'. There, the user will need to attach the 'access_token' and the 'JWT' as seen below.

Note: the only difference between PATCH and PUT is that PATCH requires only one if the entity attributes in the request body, PUT requires **all** of the entity attributes in the request body.

Path Parameters

None

Path Headers

'token' = {access_token}

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

JSON

Request JSON Attributes - see note in prerequisites

Name	Type	Description	Required?
agency	String	Name of current real estate agency	Yes

Request Body Example

```
{
  "agency": "lighthouse",
  "lname": "Frey"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing or adding more required attributes
Failure	401 Unauthorized	User has not been authorized through google
Failure	403 Forbidden	User already has an account
Failure	406 Accept type	Server does not support the requested media
Failure	415 unsupported media	The body payload was not labeled JSON

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 201 Created
{
  "UID": "101062042114979755562",
  "agency": "lighthosue",
  "fname": "Chris",
  "id": 5138716758638592,
  "lname": "Frey",
  "self": "https://final-rest-chris.wn.r.appspot.com/user/5138716758638592"
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 403 Forbidden
{
  "Error": "You are forbidden"
}
```

Failure

```
Status: 406 bad media
{
  "Error": "Server expects application/json Content-Type"
}
```

Failure

```
Status: 415 unsupported media
{
  "Error": "Not an acceptable media type"
}
```

Create a Listing

Allows you to create a new listing attributed to the user.

POST /listings

Request

Prerequisites:

The user must be authenticated with google and the user must attach the Bearer Token obtained from google to the Authorization header.

Path Parameters

None

Path Headers

None

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
price	int	Price of listing	No
sqaure_feet	int	listing_properties = ['price', 'square_feet', 'bedrooms', 'bathrooms', 'keywords', 'type', 'representative']	No
bedrooms	int	Amount bedrooms in listing	No
bathrooms	int	Amount bathrooms in listing	No
keywords	List[String]	Special keywords for listing	No
Typ	String	Type of listing i.e. 'house', 'condo'	No

Request Body Example

```
{
  "price": 500000,
  "square_feet": 2050 ,
  "bedrooms": 4,
  "bathrooms": 2.5,
  "keywords": [ ],
  "type": "house"
}
```


Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	The request is missing or adding more required attributes
Failure	401 Unauthorized	User has not been authorized through google
Failure	415 unsupported media	The body payload was not labeled JSON

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 201 Created
{
  "bathrooms": 2.5,
  "bedrooms": 4,
  "id": 5163227868561408,
  "keywords": [],
  "price": 500000,
  "representative": "101062042114979755562",
  "self": "https://final-rest-chris.wn.r.appspot.com/listings/5163227868561408",
  "square_feet": 2050,
  "type": "house"
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 415 unsupported media
{
  "Error": "Not an acceptable media type"
}
```

Get all Listings OR get user Listings

Allows you to get all listings or get all listings owned by you.

GET /listings

Request

Prerequisites:

Note: to get **only** the user's listings, the request must carry the Bearer Token of the user. To get **all** listings regardless of owner, the request must not have an authorization header. The result will be a paginated list of specified listings, limited to 5 per page.

Path Parameters

None

Path Headers

None

Path Authorization

'Bearer Token' = {JWT} if only retrieving owner's listings

Request Body

None

Request Body Format

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	User has not been authorized through google
Failure	406 Accept type	Server does not support the requested media

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The `self` attribute is built upon request.

Success

Status: 200 OK

```
{
  "next": "https://final-rest-chris.wn.r.appspot.com/listings?page=1",
  "results": [
    {
      "bathrooms": 2,
      "bedrooms": 3,
      "id": 5107442887163904,
      "keywords": [],
      "price": 545000,
      "representative": "101062042114979755562",
      "self": "https://final-rest-chris.wn.r.appspot.com/listings/5107442887163904",
      "square_feet": 1400,
      "type": "house"
    },
    {
      "bathrooms": 3,
      "bedrooms": 2,
      "id": 5117579211309056,
      "keywords": [],
      "price": 699000,
      "representative": "101062042114979755562",
      "self": "https://final-rest-chris.wn.r.appspot.com/listings/5117579211309056",
      "square_feet": 2100,
      "type": "house"
    },
    {
      "bathrooms": 2,
      "bedrooms": 3,
      "id": 5158257651875840,
      "keywords": [],
      "price": 749999,
      "representative": "101062042114979755562",
      "self": "https://final-rest-chris.wn.r.appspot.com/listings/5158257651875840",
      "square_feet": 1875,
      "type": "condo"
    },
    {
      "bathrooms": 2.5,
      "bedrooms": 4,
      "id": 5163227868561408,
      "keywords": [
        "pool"
      ],
      "price": 531000,
      "representative": "101062042114979755562",
      "self": "https://final-rest-chris.wn.r.appspot.com/listings/5163227868561408",
      "square_feet": 2050,
      "type": "house"
    },
    {
      "bathrooms": 3,
      "bedrooms": 4,
      "id": 5629978607616000,
      "keywords": [],
      "price": 985000,
      "representative": "101062042114979755562",
      "self": "https://final-rest-chris.wn.r.appspot.com/listings/5629978607616000",
      "square_feet": 2350,
      "type": "house"
    }
  ]
}
```

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 406 bad media
{
  "Error": "Server expects application/json Content-Type"
}
```

Update a Listing

Allows you to update a listing owned by the user.

PATCH or PUT /listings/<id>

Request

Prerequisites:

The user must first authenticate with google. The result of a successful authentication will reroute the user to '/login'. There, the user will need to attach the 'JWT' as seen below.

Note: the only difference between PATCH and PUT is that PATCH requires only one if the entity attributes in the request body, PUT requires **all** of the entity attributes in the request body.

Path Parameters

Listing id

Path Headers

None

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

JSON

Request JSON Attributes - see note in prerequisites

Name	Type	Description	Required?
price	int	Price of listing	No
sqare_feet	int	listing_properties = ['price', 'square_feet', 'bedrooms', 'bathrooms', 'keywords', 'type', 'representative']	No
bedrooms	int	Amount bedrooms in listing	No
bathrooms	int	Amount bathrooms in listing	No
keywords	List[String]	Special keywords for listing	No
Typ	String	Type of listing i.e. 'house', 'condo'	No

Request Body Example

```
{
  "price": 531000,
  "keywords": ["pool"]
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing or adding more required attributes
Failure	401 Unauthorized	User has not been authorized through google
Failure	403 Forbidden	User doesn't own this resource
Failure	404 Not Found	"Error": "No listing with that id exists"
Failure	406 Accept type	Server does not support the requested media
Failure	415 unsupported media	The body payload was not labeled JSON

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 200 OK
{
  "bathrooms": 2.5,
  "bedrooms": 4,
  "id": 5163227868561408,
  "keywords": [
    "pool"
  ],
  "price": 531000,
  "representative": "101062042114979755562",
  "self": "https://final-rest-chris.wn.r.appspot.com/listings/5163227868561408",
  "square_feet": 2050,
  "type": "house"
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 403 Forbidden
{
  "Error": "You are forbidden"
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No listing with that id exists"
}
```

Failure

```
Status: 406 bad media
{
  "Error": "Server expects application/json Content-Type"
}
```

Failure

```
Status: 415 unsupported media
{
  "Error": "Not an acceptable media type"
}
```

Delete a Listing

Remove listing. Listing is also removed from any user's favorites.

```
DELETE /listings/<id>
```

Request

Prerequisites:

The user must first authenticate with google. The result of a successful authentication will reroute the user to '/login'. There, the user will need to attach the 'JWT' as seen below.

Path Parameters

Listing id

Path Headers

None

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

None

Response

Response Body Format

None

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Successful
Failure	401 Unauthorized	User has not been authorized through google
Failure	403 Forbidden	User doesn't own this resource
Failure	404 Not Found	"Error": "No listing with that id exists"

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 204 No Content
```


Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 403 Forbidden
{
  "Error": "You are forbidden"
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No listing with that id exists"
}
```

Add listing to favorites

Allows you to add a listing to your favorites

```
POST /favorites/<id>
```

Request

Prerequisites:

The user must be authenticated with google and the user must attach the Bearer Token obtained from google to the Authorization header.

Path Parameters

Listing id

Path Headers

None

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
LID	int	id of the desired listing to favorite	Yes
comments	String	Comments to note about the listing	No
Strength	int	A strength of the desired listing i.e. 6/10	No

Request Body Example

```
{
  "LID": 5163227868561408,
  "comments": "Beth and Jim really liked this one",
  "strength": 7
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201	
Failure	400 Bad Request	The request is missing or adding more required attributes
Failure	401 Unauthorized	User has not been authorized through google
Failure	403 Forbidden	User doesn't own this resource
Failure	404 Not Found	"Error": "No listing with that id exists"
Failure	406 Accept type	Server does not support the requested media
Failure	415 unsupported media	The body payload was not labeled JSON

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 201 Created
{
  "LID": 5163227868561408,
  "UID": "101062042114979755562",
  "comments": "Beth and Jim really liked this one",
  "self": "https://final-rest-chris.wn.r.appspot.com/listings/5163227868561408",
  "strength": 7
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 403 Forbidden
{
  "Error": "You are forbidden"
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No listing with that id exists"
}
```

Failure

```
Status: 406 bad media
{
  "Error": "Server expects application/json Content-Type"
}
```

Failure

```
Status: 415 unsupported media
{
  "Error": "Not an acceptable media type"
}
```

Get Favorites

Allows you to get all of the user's favorites.

GET /favorites

Request

Prerequisites:

Note: The request must carry the Bearer Token of the user in the authorization header. The result will be a paginated list of specified listings, limited to 5 per page.

Path Parameters

None

Path Headers

None

Path Authorization

'Bearer Token' = {JWT} if only retrieving owner's listings

Request Body

None

Request Body Format

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	User has not been authorized through google
Failure	406 Accept type	Server does not support the requested media

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The `self` attribute is built upon request.

Success

```
Status: 200 OK
{
  "results": [
    {
      "LID": 5163227868561408,
      "UID": "101062042114979755562",
      "comments": "Beth and Jim really liked this one",
      "self": "https://final-rest-chris.wn.r.appspot.com/listings/5163227868561408",
      "strength": 7
    },
    {
      "LID": 5163227868561408,
      "UID": "101062042114979755562",
      "comments": "Beth and Jim really liked this one",
      "self": "https://final-rest-chris.wn.r.appspot.com/listings/5163227868561408",
      "strength": 7
    }
  ]
}
```

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 406 bad media
{
  "Error": "Server expects application/json Content-Type"
}
```

Update a Favorite

Updates the favorite.

PATCH or PUT /favorites/<id>

Request

Prerequisites:

The user must first authenticate with google. The result of a successful authentication will reroute the user to '/login'. There, the user will need to attach the 'JWT' as seen below.

Note: the only difference between PATCH and PUT is that PATCH requires only one if the entity attributes in the request body, PUT requires **all** of the entity attributes in the request body.

Path Parameters

Listing id

Path Headers

None

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

JSON

Request JSON Attributes - see note in prerequisites

Name	Type	Description	Required?
LID	int	id of the desired listing to favorite	Yes
comments	String	Comments to note about the listing	No
Strength	int	A strength of the desired listing i.e. 6/10	No

Request Body Example

```
{
  "LID": 5163227868561408,
  "comments": "Beth and Jim thought this was OK",
  "strength": 5
}
```

Response

Response Body Format

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The request is missing or adding more required attributes
Failure	401 Unauthorized	User has not been authorized through google
Failure	403 Forbidden	User doesn't own this resource
Failure	404 Not Found	"Error": "No listing with that id exists"
Failure	406 Accept type	Server does not support the requested media
Failure	415 unsupported media	The body payload was not labeled JSON

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 200 Created
{
  "LID": 5163227868561408,
  "UID": "101062042114979755562",
  "comments": "Beth and Jim thought this was OK",
  "self": "https://final-rest-chris.wn.r.appspot.com/listings/5163227868561408",
  "strength": 5
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 403 Forbidden
{
  "Error": "You are forbidden"
}
```

Failure


```
Status: 404 Not Found
{
  "Error": "No favorite with that id exists"
}
```

Failure

```
Status: 406 bad media
{
  "Error": "Server expects application/json Content-Type"
}
```

Failure

```
Status: 415 unsupported media
{
  "Error": "Not an acceptable media type"
}
```

Delete a Favorite

Removes listing from the user's favorites.

```
DELETE /favorites/<id>
```

Request

Prerequisites:

The user must first authenticate with google. The result of a successful authentication will reroute the user to '/login'. There, the user will need to attach the 'JWT' as seen below.

Path Parameters

Listing id

Path Headers

None

Path Authorization

'Bearer Token' = {JWT}

Request Body

Required

Request Body Format

None

Response

Response Body Format

None

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Successful
Failure	401 Unauthorized	User has not been authorized through google
Failure	404 Not Found	"Error": "No favorite with that listing id exists"

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The self attribute is built upon request.

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
{
  "Error": "You are not authorized"
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No listing with that id exists"
}
```