

Final Project API Design

App Description

This app is designed for realtors who wish to keep information about houses on the market. After receiving an account, the realtor can choose to track a house, update information about it, and un-track a house. At any point in time, any of the realtors can check which houses are being tracked by any other realtor. A house can only be tracked by one realtor at a time; this keeps the market competitive.

Open houses are very important to realtors since it is an opportunity for them to meet potential clients. Participation at open houses is heavily dependent on the weather. Because of this, realtors would like to know what the weather is like at a current location at any point in time. Weather information can be displayed about the house using a real-time, external 3rd party weather API service. The information is displayed is a 5-day forecast with data every three hours for a particular ZIP code. This helps the realtor to be informed about where they need to spend their time. Because only 60 calls can be made per minute to the API service, we are limited to only requesting weather information when requesting single houses. The third party service used is OpenWeatherMap: <https://openweathermap.org/api>.

App authorization

This app is authorized using Google's OAuth2 API. To demonstrate this, two personal user accounts were used to receive authorizations tokens from Google. Along with the tokens came a refresh token. These refresh tokens are required for each request that is made where user specific information is needed. See the [link](#) to the Piazza discussion on this for more information.

Other notes

All response bodies are to be sent as 'application/json', which is to say the body is to be written in JSON format. The application is to be run on localhost, since a live public URL is not required (see [link](#) to Piazza discussion).

Important note for running Postman tests:

When running the POSTMAN test runner collection, make sure that there is a 100ms delay between each API call. This is to deal with eventual consistency for the data on the server.

Get all users

GET /users

Response

Status: 200 OK

Get a single user

GET /users/{id}

Header Request Data

Authorization: {token}

Note: If the token passed matches the user that is being looked up, the user's Google+ email and Google ID will also be displayed

Responses

Status: 200 Ok

Status: 404 Not Found

Occurs when an invalid user ID is provided

Creating a user

POST /user

Only admins can create users

Header Request Data

Authorization: {token}

Body Request Data

gid (string) – required, the google ID of the user

age (number) – required

hairColor (string) – required

isAdmin (boolean) – optional, default false

Responses

Status: 201 Created

Status: 400 Bad Request

Occurs when one or more of the following is true:

- One or more of the required fields is not provided
- One or more of the required fields uses the wrong data type

Status: 403 Forbidden

Occurs when the token does not belong to a user that is an admin

Notes

All created users are by default, not admins, and created users are given an automatically generated ID

Modifying a User

PATCH /users/{id}

Header Request Data

Authorization: {token}

Note: The authorization token must be from a user that is an admin or must belong to the user {id} in the route

Body Request Data

hairColor (string) – optional

age (number) – optional

isAdmin (number) – optional, can only be set if the user is an admin

Responses

Status: 204 No Content

Status: 400 Bad Request

Occurs when one or more of the following is true:

- No request body is provided
- One or more of the parameters uses the wrong data type

Status: 403 Forbidden

Occurs when a non-admin user attempts to edit someone else

Deleting a User

DELETE /users /{id}

Only admins can use this command

Header Request Data

Authorization: {token}

Responses

Status: 204 No Content

Status: 400 Bad Request

Occurs when an invalid id is provided

Status: 403 Forbidden

Occurs when the user making the API call is not an admin

Get all Houses

GET /houses

Response

Status: 200 Ok

Get all Your Tracked Houses

GET /houses

Header Request Data

Authorization: {token}

Response

Status: 200 Ok

Status: 400 Bad Request

Occurs when the token is invalid

Notes

The token belongs to the user who is tracking the houses

Get a Single House

GET /houses/{id}

Responses

Status: 200 Ok

Status: 404 Not Found

Occurs when the supplied ID is invalid

Track a house

POST /houses

Header Request Data

Authorization: {token}

Body Request Data

address (string) – required

price (string) – required

zip (string) – required

Note: the address and ZIP combination must be unique

Responses

Status: 201 Created

Status: 400 Bad Request

Occurs when one or more of the following occurs:

- A body request parameter is not present
- A parameter uses the wrong data type
- The address and ZIP combination are not unique
- The user is invalid

Notes

The house is automatically tracked by the user who makes the request

Modifying House Details

PATCH /houses/{id}

Header Request Data

Authorization: {token}

Body Request Data

price (string) – optional

Responses

Status: 204 No Content

Status: 403 Forbidden

Occurs when the token does not belong to the realtor tracking the house or if the token is invalid.

Status: 400 Bad Request

Occurs when one or more of the following is true:

- No request body is passed
- One or more of the parameters uses an incorrect data type

Replacing a House

PUT /houses/{id}

Header Request Data

Authorization: {token}

Body Request Data

address (string) – required

price (string) – required

zip (string) - required

Responses

Status: 201 Created

Status: 403 Forbidden

Occurs when the token does not belong to the realtor tracking the house previous house or an invalid token was used.

Status: 400 Bad Request

Occurs when one or more of the following is true:

- The number provided uses an incorrect data type
- One or more of the response body parameters is missing
- The address and zip combination are already taken

Notes

This will replace the old ID with a new house ID

Un-tracking a House

DELETE /houses/{id}

Header Request Data

Authorization: {token}

Responses

Status: 204 No Content

Status: 403 Forbidden

Occurs when the token does not belong to the realtor tracking the house previous house or an invalid token was used.

Status: 404 Not Found

Occurs when an invalid house ID is provided

Notes

Admins can un-track any house for any user

Look up 5-day weather forecast about a house

GET /houses/{id}/forecast

Responses

Status: 200 Ok

Status: 404 Not Found

Occurs if the ID is invalid

Status: 400 Bad Request

Occurs if there is an error with the 3rd party API