

Darryl Nichols

Back End Development Final Project – Real estate Offer and Investing Service

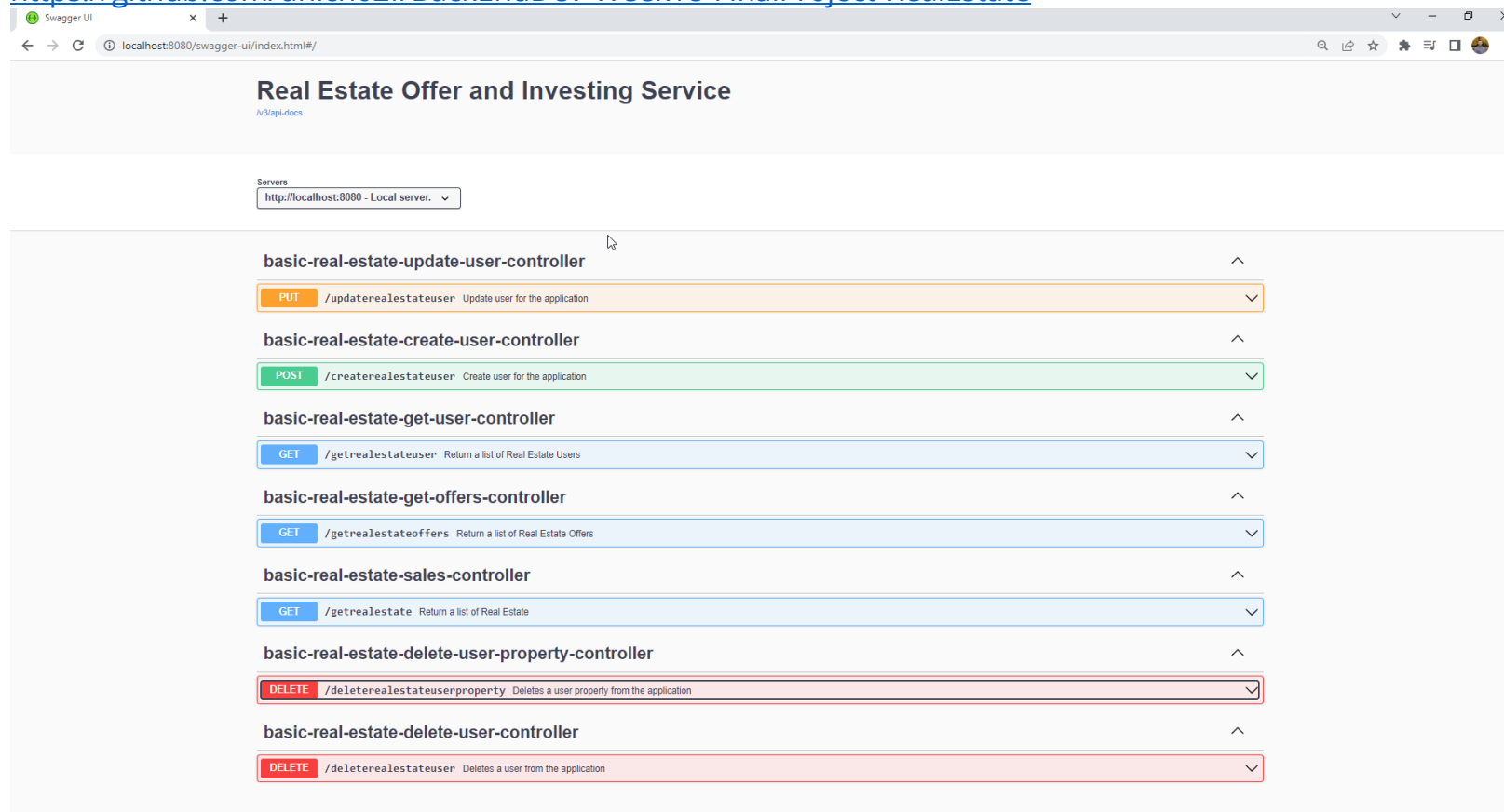
Week 18

09/01/2022

Create a Final Project PDF

- Include the GitHub Repo URL of your Project

<https://github.com/dnich02f/BackEndDev-Week18-FinalProject-RealEstate>



- Include Screenshots of the RESULTS of RUNNING each path in your WebAPI working:
  - One entity should have 4 CRUD operations: USER
    - GET a User – DONE
      - Allowed Requests: Shown in “Requested URL”
      - Input Required: FirstName, LastName as Strings as shown in the “Parameters”
      - Path Variables: firstName, LastName, as shown in the “Curl”
      - Expected Outcome: Return a list of real estate users as shown in the “Example Value” under “Reponses, Code 200”.
      - HTTPStatuses: 200, 400, 404, 500

Swagger UI

localhost:8080/swagger-ui/index.html#/basic-real-estate-get-user-controller/fetchUsers

GET /getRealEstateUser Returns a list of Real Estate Users

Parameters

Name Description

first\_name \*required Enter desired user's first name, (i.e., "First")

string (query)

last\_name \*required Enter desired user's last name, (i.e., "Last")

string (query)

Execute Clear

Responses

200

Content

HTTP/1.1 200 OK (application/json)

Request URL

http://localhost:8080/getRealEstateUser?first\_name=LeBron&last\_name=James

Server response

Code Details

200

Response body

```
{
  "users": [
    {
      "first_name": "LeBron",
      "last_name": "James",
      "phone": "555-555-5555"
    }
  ]
}
```

Response headers

connection: keep-alive  
content-type: application/json  
date: Sat, 31 Aug 2019 13:23:20 GMT  
keep-alive: timeout=60  
transfer-encoding: chunked

200

A list of Real Estate Users is returned

Media type: application/json

Example Value

```
{
  "users": [
    {
      "first_name": "LeBron",
      "last_name": "James",
      "phone": "555-555-5555"
    }
  ]
}
```

400

The request parameters are invalid

Media type: application/json

404

No Real Estate Users were found with the input criteria

Media type: application/json

500

An unplanned error occurred

Media type: application/json

DBeaver 22.1.3 - <realestate> Script

File Edit Navigate Search SQL Editor Database Window Help

Auto

realestate realestate

Enter a part of object name here

jeep - localhost:3306

realestate - localhost:3306

Databases

realestate

Tables

Views

Indexes

Procedures

Triggers

Events

Users

Administrator

System Info

select \* from users u

Output

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'INSERT INTO use'

users 1

select \* from users u

Grid

	user_pk	first_name	last_name	phone
1	1	Hektor	Attaway	755.223.5969
2	2	Torbjorg	Maynard	256.399.4665
3	3	Agni	Van Altena	[NULL]
4	4	Terzo	Kappel	328.993.3774
5	5	Businge	Foster	399.377.3854
6	6	LeBron	James	555-555-5550
7	9	FIRST	LAST	555-555-1234

Text

Record

Save Cancel Script

200 7 Rows: 1

7 row(s) fetched - 1ms, on Aug 31, 13:23:20

MST en Writable Smart Insert

## CREATE a User – DONE

- Allowed Requests: Shown in “Requested URL”
- Input Required: FirstName, LastName, phone as Strings as shown in the “Parameters”
- Path Variables: firstName, LastName, phone as shown in the “Curl”
- Expected Outcome: Returns the created real estate user as shown in the “Example Value” under “Responses, Code 201”.
- HTTPStatuses: 201, 400, 404, 500

The image displays two side-by-side application windows. The left window is the Swagger UI for a REST API, showing the 'POST /createRealEstateUser' endpoint. The parameters section lists 'firstName' (string, required), 'lastName' (string, required), and 'phone' (string, required). The response section shows a 201 status code with a JSON body representing a user object. The right window is DBeaver 22.1.3, showing a SQL query 'select \* from users u' executed against a database named 'realstate'. The query results are displayed in a table with 8 rows, including columns for 'user\_pk', 'first\_name', 'last\_name', and 'phone'.

**Swagger UI Parameters:**

Name	Description
firstName	string (required)
lastName	string (required)
phone	string (required)

**Swagger UI Response (201):**

```
{  "user_pk": 1,  "first_name": "Hektor",  "last_name": "Attaway",  "phone": "755-223-5969"}
```

**DBeaver SQL Query:**

```
select * from users u
```

**DBeaver Query Results:**

user_pk	first_name	last_name	phone
1	Hektor	Attaway	755-223-5969
2	Torbjerg	Maynard	256-399-4665
3	Agni	Van Altena	[NULL]
4	Terzo	Kappel	328-993-3774
5	Businge	Foster	399-377-3854
6	LeBron	James	555-555-5550
7	FIRST	LAST	555-555-1234
11	Average	Joe	555-555-3579

- DELETE a User – DONE
  - Allowed Requests: Shown in “Requested URL”
  - Input Required: userPK as an int as shown in the “Parameters”
  - Path Variables: userPK as shown in the “Curl”
  - Expected Outcome: Deletes real estate user with the userPk as shown in the “Example Value” under “Responses, Code 201”.
- HTTPStatuses: 201, 400, 404, 500

The image displays two software interfaces side-by-side. The left interface is Swagger UI, showing the documentation for the 'DELETE /deleteRealEstateUser' endpoint. The right interface is DBeaver, showing a SQL query and the resulting data from the 'users' table.

**Swagger UI Details:**

- Endpoint:** DELETE /deleteRealEstateUser
- Description:** Deletes a user from the application
- Parameters:**
  - Name:** UserPK
  - Value:** 9
- Responses:**
  - 201:** The user has been deleted. Example Value: { "application/json": {} }
  - 400:** The request parameters are invalid. Example Value: { "application/json": { "message": "Invalid request parameters" } }
  - 404:** A user with the parameter was not found. Example Value: { "application/json": { "message": "User not found" } }
  - 500:** An unplaned error occurred. Example Value: { "application/json": { "message": "Internal server error" } }

**DBeaver Details:**

- SQL Query:** select \* from users u
- Table:** users
- Data:**

id	user_pk	first_name	last_name	phone
1	Hektor	Attaway	755.223.5969	
2	Torbjerg	Maynard	256.399.4665	
3	Agni	Van Altena	[NULL]	
4	Terzo	Kappel	328.993.3774	
5	Businge	Foster	399.377.3854	
6	LeBron	James	555-555-5550	

- UPDATE a User – DONE
  - Allowed Requests: Shown in “Requested URL”
  - Input Required: userPk, FirstName, LastName, phone as Strings as shown in the “Parameters”
  - Path Variables: userPk, firstName, LastName, phone as shown in the “Curl”
  - Expected Outcome: Returns the updated real estate user as shown in the “Example Value” under “Reponses, Code 201”.
- HTTPStatuses: 201, 400, 404, 500

Swagger UI interface for the `updateUser` endpoint. The interface shows the following details:

- Method:** PUT
- Path:** /updateUser
- Parameters:**
  - `userPk` (string, required): Input field with value 5.
  - `firstName` (string, required): Input field with value Adrian.
  - `lastName` (string, required): Input field with value Foster.
  - `phone` (string, required): Input field with value 399-377-3854.
- Responses:**
  - 201:** The user's information was updated. Example value: `{ "userPk": 5, "firstName": "Adrian", "lastName": "Foster", "phone": "399-377-3854" }`.
  - 400:** The request parameters are invalid.
  - 404:** A user was not found with the input criteria.
  - 500:** An unhandled error occurred.

DBeaver 22.1.3 interface showing a SQL query and a table view of the `users` table.

**SQL Query:** `select * from users u`

**Table View:**

user_pk	first_name	last_name	phone
1	Hektor	Attaway	755-223-5969
2	Torbjerg	Maynard	256-399-4665
3	Agni	Van Altena	[NULL]
4	Terzo	Kappel	328-993-3774
5	Adrian	Foster	399-377-3854
6	LeBron	James	555-555-5550

6 row(s) fetched - 1ms, on Aug 31, 13:28:46

- Three other entities should have a CRUD operation:
  - GET Properties – DONE
    - Allowed Requests: Shown in “Requested URL”
    - Input Required: propertyType and propertyStatus as enum options, as shown in the “Parameters”
    - Path Variables: propertyType and propertyStatus as shown in the “Curl”
    - Expected Outcome: Returns real estate properties as shown in the “Example Value” under “Reponses, Code 200”.
  - HTTPStatuses: 200, 400, 404, 500

The image shows two software interfaces side-by-side. On the left is the Swagger UI for a REST API, and on the right is the DBeaver database client.

**Swagger UI (Left):** The browser window shows the Swagger UI for the endpoint `basic-real-estate-get-offers-controller`. It displays the API documentation for the `basic-real-estate-get-property-controller` endpoint. The parameters section shows two required parameters: `propertyType` (enum) and `propertyStatus` (enum). The response section shows a 200 status code with an example JSON response.

**DBeaver (Right):** The DBeaver 22.1.3 interface shows a SQL query in the SQL Editor: `select * from properties p`. The query results are displayed in a table with 11 rows. The table has columns: `property_pk`, `property_id`, `square_feet`, `bedrooms`, `bathrooms`, `year_built`, `price`, `rental_income`, and `status`.

property_pk	property_id	square_feet	bedrooms	bathrooms	year_built	price	rental_income	status
1	1	3,480	5	2.5	1,952	499,900	2,100	A
2	2	1,974	2	1	1,916	265,000	1,430	A
3	3	2,010	5	2	2,002	349,000	1,900	A
4	4	2,578	4	3	1,917	355,000	1,300	A
5	5	7,183	5	6	2,008	1,275,000	4,300	A
6	6	3,600	6	3	2,008	470,000	2,800	A
7	7	2,996	5	3	1,955	599,500	2,950	A
8	8	2,010	5	3	2,002	349,000	2,300	A
9	9	1,882	4	2	1,950	264,000	1,900	A
10	10	80,000	80	60	2,010	4,500,000	100,000	A
11	11	16,000	16	8	2,002	1,350,000	18,000	A

- DELETE User\_Properties – DONE
  - Allowed Requests: Shown in “Requested URL”
  - Input Required: propertyPK as an int as shown in the “Parameters”
  - Path Variables: propertyPK as shown in the “Curl”
  - Expected Outcome: Deletes real estate user property connection with the propertyPk as shown in the “Example Value” under “Reponses, Code 201”.
- HTTPStatuses: 201, 400, 404, 500

## BEFORE

The image shows two side-by-side screenshots. The left screenshot is from the Swagger UI, displaying the API documentation for the 'basic-real-estate-delete-user-property-controller'. The 'DELETE /deleteRealEstateUserProperty' endpoint is highlighted, showing a required parameter 'propertyPK' of type 'integer (32bit)'. The 'Responses' section lists status codes 201, 400, 404, and 500 with their respective descriptions. The right screenshot is from DBeaver, showing a SQL query 'select \* from user\_property up' in the SQL Editor. The 'user\_property' table is visible in the 'realstate' database. The 'Grid' tab shows two rows of data. The 'Text' tab shows an error message: 'You have an error in your INSERT INTO user' at line 1'.

**Swagger UI Screenshot:**

**basic-real-estate-delete-user-property-controller**

**DELETE** /deleteRealEstateUserProperty Deletes a user property from the application

Deletes a user property from the application

**Parameters**

Name	Description
propertyPK	required integer (32bit) (query)

**Responses**

Code	Description	Links
201	The user property has been deleted	API 201s
400	The request parameters are invalid	API 400s
404	A user property with the parameter was not found	API 404s
500	An unplanned error occurred	API 500s

**DBeaver Screenshot:**

**realstate**

**user\_property**

**SQL Editor:**

```
select * from user_property up
```

**Grid:**

	user_fk	property_fk
1	1	11
2	1	10

**Text:**

You have an error in your INSERT INTO user' at line 1



# AFTER

The Swagger UI displays the DELETE endpoint for the `basic-real-estate-delete-user-property-controller`. The endpoint is `/deleteRealEstateUserProperty` and it requires a `propertyPK` parameter of type `integer (int32)` with a value of `10`. The response status is `200` (OK) with a description "The user property has been deleted". The response body is a JSON object: `{ "message": "The user property has been deleted" }`. The UI also shows the request and response headers and the server response.

The DBeaver 22.1.3 - `<realestate> Script` window shows a SQL query: `select * from user_property up`. The query is executed, and the results are displayed in a table with 1 row(s) fetched. The table has columns `user_fk` and `property_fk`. The status bar shows "1 row(s) fetched - 1ms, on Aug 31, 13:52:03".

An error message is displayed in the Output window: "You have an error in your INSERT INTO user' at line".

- GET Offers – DONE
  - Allowed Requests: Shown in “Requested URL”
  - Input Required: None. It returns all offers the “Parameters”
  - Path Variables: None. as shown in the “Curl”
  - Expected Outcome: Return a list of real estate offers as shown in the “Example Value” under “Responses, Code 200”.
- HTTPStatuses: 200, 400, 404, 500

The image displays two side-by-side application windows. The left window is the Swagger UI for a REST API, and the right window is the DBeaver database client.

**Swagger UI (Left Window):**

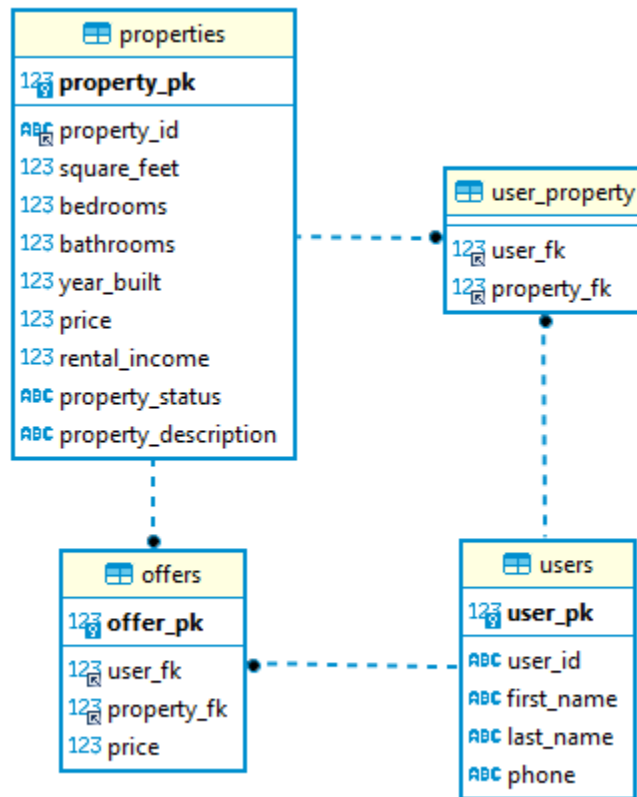
- URL:** `localhost:8080/swagger-ui/index.html#/basic-real-estate-get-offers-controller/fetchOffer`
- Endpoint:** `GET /getrealstateoffers` (Returns a list of Real Estate Offers)
- Parameters:** None.
- Responses:**
  - 200:** Returns a list of Real Estate Offers. The response body is a JSON array of objects, each containing fields like `offer_pk`, `user_fk`, `property_fk`, and `price`.
  - 400:** The request parameters are invalid.
  - 404:** No Real Estate Offers were found with the input criteria.
  - 500:** An unplanned error occurred.

**DBeaver (Right Window):**

- Script:** `select * from offers o`
- Output:** You have an error in your INSERT INTO user' at line
- Table:** `offers` (1 row selected)
- Columns:** `offer_pk`, `user_fk`, `property_fk`, `price`
- Data:**

offer_pk	user_fk	property_fk	price
1	6	10	4,250,000
2	1	10	4,200,000

- Include your Database Design ERD!



- Upload PDF to File Submission in the LMS for **Week 18 Final Project** under *Week 18 Final Project Submission*