



Etuloc API 1.0 OAS 3.0

Etuloc is a platform designed to streamline the renting process between university students and property owners. The app aims to simplify house-hunting and rental management, helping students find housing options easily and allowing property owners to manage their listings efficiently.

Authorize

user

POST /user/student Add a new student

Creates a new student user in the system.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "phoneNumber": "string",
  "student": true,
  "university": "string",
  "sumOfRatings": 0,
  "numberOfRaters": 0,
  "city": "string",
  "rating": "string",
  "postedHouses": [
    "string"
  ],
  "numberRentedHouses": 0
}
```

Responses

| Code | Description | Links |
|------|-----------------------------|----------|
| 201 | Student successfully added. | No links |
| 500 | Internal server error. | No links |

GET /user/student Get all students



Fetch all students from the system.

Parameters

Try it out

No parameters

Responses

| Code | Description | Links |
|------|--------------------------------|----------|
| 200 | Students fetched successfully. | No links |

Media type

application/json

Controls Accept header.

Example Value Schema

```
[
  {
    "email": "student1@example.com",
    "name": "Alice",
    "student": true,
    "university": "University A",
    "sumOfRatings": 3,
    "city": "Chicago",
    "postedHouses": [],
    "numberRentedHouses": 0
  },
  {
    "email": "student2@example.com",
    "name": "Bob",
    "student": true,
    "university": "University B",
    "sumOfRatings": 8,
    "city": "San Francisco",
    "postedHouses": [
```

| Code | Description | Links |
|------|---|----------|
| | <pre> "House A"], "numberRentedHouses": 1 }]</pre> | |
| 500 | Internal server error. | No links |

POST

/user/Owner

Add a new owner

^

Creates a new owner user in the system.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "name": "string",
  "email": "string",
  "password": "string",
  "phoneNumber": "string",
  "student": true,
  "university": "string",
  "sumOfRatings": 0,
  "numberOfRaters": 0,
  "city": "string",
  "rating": "string",
  "postedHouses": [
    "string"
  ],
  "numberRentedHouses": 0
}
```

Responses

| Code | Description | Links |
|------|---------------------------|----------|
| 201 | Owner successfully added. | No links |
| 500 | Internal server error. | No links |

GET

/user

Get all users

^

Fetches all users (students and owners) in the system.

Parameters

Try it out

No parameters

Responses

| Code | Description | Links |
|------|---|----------|
| 200 | <div>Users fetched successfully.</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "email": "owner@example.com", "name": "John Doe", "student": false, "university": "N/A", "sumOfRatings": 10, "city": "New York", "postedHouses": ["House 1", "House 2"], "numberRentedHouses": 2 }, { "email": "student@example.com", "name": "Jane Doe", "student": true, "university": "University XYZ", "sumOfRatings": 5, "city": "Los Angeles", "postedHouses": [], "numberRentedHouses": 0 }]</div> | No links |
| 500 | Internal server error. | No links |

GET

/user/{id}

Get user by ID

^

Fetch a specific user by their ID.

Parameters

Try it out

| Name | Description |
|--|---------------|
| id * required string (path) | <div>id</div> |

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | <p>User fetched successfully.</p> <p>Media type</p> <div>application/json</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "email": "student@example.com", "name": "Jane Doe", "student": true, "university": "University XYZ", "sumOfRatings": 5, "city": "Los Angeles", "postedHouses": [], "numberRentedHouses": 0 }</pre> | No links |
| 404 | User not found. | No links |

PATCH /user/{id} Update user details



Update the details of an existing user (student or owner).

Parameters

Try it out

Name

Description

id * required

string (path)

id

Request body required

application/json

Example Value

Schema

```
{  "name": "string",  "email": "string",  "password": "string",  "phoneNumber": "string",  "student": true,  "university": "string",  "sumOfRatings": 0,  "numberOfRaters": 0,  "city": "string",  "rating": "string",  "postedHouses": [    "string"  ],  "numberRentedHouses": 0}
```

Responses

| Code | Description | Links |
|------|---------------------------------------|----------|
| 200 | User updated successfully. | No links |
| 400 | Bad Request, invalid data or user ID. | No links |

DELETE

/user/{id}

Delete user by ID

^

Deletes a user (student or owner) by their ID.

Parameters

Try it out

| Name | Description |
|--|---------------|
| id * required string (path) | <div>id</div> |

Responses

| Code | Description | Links |
|------|----------------------------|----------|
| 200 | User deleted successfully. | No links |
| 404 | User not found. | No links |

auth

^

POST /auth/login User login ^

Allows a registered user to log in by providing their email and password.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

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

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | Login successful, returns a JWT token. | No links |
| 401 | Unauthorized, invalid credentials. | No links |

POST

/auth/register

User registration

^

Registers a new user with their email, password, and other necessary details.

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{  "name": "string",  "email": "string",  "password": "string",  "phoneNumber": "string",  "student": true,  "university": "string",  "sumOfRatings": 0,  "numberOfRaters": 0,  "city": "string",  "rating": "string",  "postedHouses": [    "string"  ],  "numberRentedHouses": 0}
```

Responses

| Code | Description | Links |
|------|--|----------|
| 201 | User successfully registered. | No links |
| 400 | Bad Request, validation errors or user already exists. | No links |

house



POST

/house

Create a new house

^

Parameters

Try it out

No parameters

Request body

required

application/json

Example Value

Schema

```
{
  "email": "string",
  "price": 0,
  "images": [
    "string"
  ],
  "gender": "string",
  "furniture": true,
  "roomsNumber": 0,
  "city": "string"
}
```

Responses

| Code | Description | Links |
|------|---|-------|
| 201 | <div>House created successfully.</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div><pre>{ "email": "string", "price": 0, "images": ["string"], "gender": "string", "furniture": true, "roomsNumber": 0, "city": "string" }</pre></div> | |

 No links || 500 | Internal Server Error | No links |

GET

/house

Get all houses

^

Parameters

Try it out

No parameters

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | <div>Houses fetched successfully.</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "email": "string", "price": 0, "images": ["string"], "gender": "string", "furniture": true, "roomsNumber": 0, "city": "string" }]</div> | No links |

GET /house/{id} Get a specific house by ID



Parameters

Try it out

Name Description

id * required
string
(path)

id

Responses

| Code | Description | Links |
|------|-----------------------------|----------|
| 200 | House fetched successfully. | No links |

| Code | Description | Links |
|------|---|----------|
| | <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "email": "string", "price": 0, "images": ["string"], "gender": "string", "furniture": true, "roomsNumber": 0, "city": "string"}</pre></div> | |
| 404 | House not found | No links |

PATCH

/house/{id} Update an existing house

Parameters

Try it out

| Name | Description |
|--|---------------|
| <div>id * required</div> <div>string</div> <div>(path)</div> | <div>id</div> |

Request body required

application/json

Example Value Schema

```
{  "email": "string",  "price": 0,  "images": [    "string"  ],  "gender": "string",  "furniture": true,  "roomsNumber": 0,  "city": "string"}
```

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | <div>House updated successfully.</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "email": "string", "price": 0, "images": ["string"], "gender": "string", "furniture": true, "roomsNumber": 0, "city": "string"}</pre></div> | No links |
| 400 | Bad Request | No links |

GET

/house/renter/{email}

Get houses for a specific renter by email

^

Parameters

Try it out

| Name | Description |
|---|------------------|
| <div>email * required</div> <div>string</div> <div>(path)</div> | <div>email</div> |

Responses

| Code | Description | Links |
|------|---|----------|
| 200 | <div>Houses loaded successfully.</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "email": "string", "price": 0,</pre></div> | No links |

| Code | Description | Links |
|------|--|-------|
| | <pre>"images": ["string"], "gender": "string", "furniture": true, "roomsNumber": 0, "city": "string" }]</pre> | |

Schemas

CreateUserDto

{
 name*
 string
 The full name of the user

 email*
 string
 The email address of the user

 password*
 string
 The password of the user, must be strong

 phoneNumber*
 string
 The phone number of the user

 student*
 boolean
 Indicates if the user is a student or an owner

 university*
 string
 The university the user attends, applicable if the user is a student

 sumOfRatings*
 number
 Total sum of ratings given to the owner by students

 numberOfRaters*
 number
 the number of the students that rated a specific owner

 city*
 string
 The city where the user resides or operates

 rating*
 string
 The rate of an owner

 postedHouses*
 [
 List of houses posted by the user (if the user is an owner)
]

 numberRentedHouses*
 string]
 number
 The number of houses rented by the user (if the user is an owner)
}

```
UpdateUserDto {
  name
      string
      The full name of the user

  email
      string
      The email address of the user

  password
      string
      The password of the user, must be strong

  phoneNumber
      string
      The phone number of the user

  student
      boolean
      Indicates if the user is a student or an owner

  university
      string
      The university the user attends, applicable if the user is a
      student

  sumOfRatings
      number
      Total sum of ratings given to the owner by students

  numberOfRaters
      number
      the number of the students that rated a specific owner

  city
      string
      The city where the user resides or operates

  rating
      string
      The rate of an owner

  postedHouses
      [
      List of houses posted by the user (if the user is an owner)

      [...]]

  numberRentedHouses
      number
      The number of houses rented by the user (if the user is an
      owner)
}
```

```
SignInDto {
  email*
      string
      The email of the user

  password*
      string
      The password of the user
}
```

```
CreateHouseDto  {
  email*          string
                  The email address of the house owner

  price*          number
                  The price of the house per month

  images*         [
                  The images of the house

                  string]

  gender*         string
                  The gender for which the house is intended (female or male)

  furniture*      boolean
                  Indicates whether the house is furnished (true if furnished,
                  false if not)

  roomsNumber*   number
                  The number of rooms in the house

  city*           string
                  The city where the house is located
}
```

```
UpdateHouseDto  {
  email          string
                  The email address of the house owner

  price          number
                  The price of the house per month

  images         [
                  The images of the house

                  string]

  gender         string
                  The gender for which the house is intended (female or male)

  furniture      boolean
                  Indicates whether the house is furnished (true if furnished,
                  false if not)

  roomsNumber   number
                  The number of rooms in the house

  city          string
                  The city where the house is located
}
```