

Endpoints

1. GET /notes

Fetch all notes in the database.

Request

- **Method:** GET
- **URL:** /notes
- **Query Parameters:** None

Response

- **Status:** 200 OK
- **Body:** A JSON array of all notes.

```
[
  {
    "id": 1,
    "title": "Note 1",
    "content": "This is the content of note 1.",
    "owner": 1,
    "created_at": "2024-11-09T00:00:00Z",
    "updated_at": "2024-11-09T00:00:00Z"
  },
  {
    "id": 2,
    "title": "Note 2",
    "content": "This is the content of note 2.",
    "owner": 2,
    "created_at": "2024-11-09T00:00:00Z",
    "updated_at": "2024-11-09T00:00:00Z"
  }
]
```

2. POST /notes

Create a new note.

Request

- **Method:** POST
- **URL:** /notes
- **Body:** A JSON object containing the note's title, content, and owner ID.

```
{
  "title": "New Note",
```

```
"content": "This is a new note.",  
"owner": 1  
}
```

Response

- **Status:** 201 Created
- **Body:** A JSON object containing the newly created note.

```
{  
  "id": 3,  
  "title": "New Note",  
  "content": "This is a new note.",  
  "owner": 1,  
  "created_at": "2024-11-09T00:00:00Z",  
  "updated_at": "2024-11-09T00:00:00Z"  
}
```

3. GET /notes/user

Fetch all notes by a specific user.

Request

- **Method:** GET
- **URL:** /notes/user
- **Query Parameters:**
 - **user_id** (required): The ID of the user whose notes you want to fetch.

```
/notes/user?user_id=1
```

Response

- **Status:** 200 OK
- **Body:** A JSON array of notes for the specified user.

```
[  
  {  
    "id": 1,  
    "title": "Note 1",  
    "content": "This is the content of note 1.",  
    "owner": 1,  
    "created_at": "2024-11-09T00:00:00Z",  
    "updated_at": "2024-11-09T00:00:00Z"  
  }  
]
```

```
}  
]
```

4. PUT /notes/update

Update an existing note.

Request

- **Method:** PUT
- **URL:** /notes/update
- **Query Parameters:**
 - **id** (required): The ID of the note to update.

```
/notes/update?id=1
```

- **Body:** A JSON object containing the updated title and content for the note.

```
{  
  "title": "Updated Note Title",  
  "content": "Updated content of the note."  
}
```

Response

- **Status:** 200 OK
- **Body:** A JSON object of the updated note.

```
{  
  "id": 1,  
  "title": "Updated Note Title",  
  "content": "Updated content of the note.",  
  "owner": 1,  
  "created_at": "2024-11-09T00:00:00Z",  
  "updated_at": "2024-11-09T00:00:00Z"  
}
```

5. DELETE /notes/delete

Delete a note by its ID.

Request

- **Method:** DELETE

- **URL:** `/notes/delete`
- **Query Parameters:**
 - `id` (required): The ID of the note to delete.

```
/notes/delete?id=1
```

Response

- **Status:** `200 OK`
- **Body:** A JSON object of the deleted note.

```
{
  "id": 1,
  "title": "Note 1",
  "content": "This is the content of note 1.",
  "owner": 1,
  "created_at": "2024-11-09T00:00:00Z",
  "updated_at": "2024-11-09T00:00:00Z"
}
```

Error Responses

The API may return the following error responses:

- **400 Bad Request:** Invalid or missing parameters.
- **404 Not Found:** The resource was not found (e.g., no note found for the provided ID).
- **405 Method Not Allowed:** The HTTP method used is not supported for the requested endpoint.
- **500 Internal Server Error:** An unexpected error occurred while processing the request.

Example Usage

Fetching all notes:

```
curl -X GET http://localhost:4321/notes
```