

Weather App: API Documentation

This is the documentation for the Weather API.

Contents

- [Registering a User](#)
- [Logging in a User](#)
- [Getting a User Profile](#)
- [Updating a User Profile](#)
- [Deleting a User Profile](#)
- [Getting a User's Credit Balance](#)
- [Purchasing User Credits](#)
- [Getting a User's Locations](#)
- [Adding a New Location](#)
- [Deleting a User's Location](#)
- [API Structure Diagram](#)

Registering a User

Request type: POST

Request body schema: application/json

API call

http://localhost:5000/api/v2/users/register

Request body

- username**
 - Type:** String
 - Required:** Yes
 - Description:** A unique username.
- email**
 - Type:** String (Email format)
 - Required:** Yes
 - Description:** A unique email address.
- password**
 - Type:** String
 - Required:** Yes
 - Description:** A unique password.

Example response

```
{
  "message": "User registered successfully."
}
```

Response fields

- message:** A simple confirmation message.

Responses

- 200 - Successful Response
 - Response schema:** application/json
 - message**
 - Type:** String
- 422 - Validation Error
 - Response schema:** application/json
 - detail**
 - detail.loc**
 - Type:** Array of string or integers
 - detail.msg**
 - Type:** String
 - detail.type**
 - Type:** String

[↑ Return to top](#)

Logging in a User

Request type: POST

Request body schema: application/x-www-form-urlencoded

API call

http://localhost:5000/api/v2/users/login

Request body

- grant_type**
 - Type:** String or null
 - Required:** No
- username**
 - Type:** String
 - Required:** Yes
 - Description:** A unique username.
- password**
 - Type:** String
 - Required:** Yes
 - Description:** A unique password.

- **scope**
 - **Type:** String (Default: "")
 - **Required:** No
- **client_id**
 - **Type:** String or null
 - **Required:** No
- **client_secret**
 - **Type:** String or null
 - **Required:** No

Example response

```
{
  "message": "User successfully logged in.",
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJzdHJpbmciLCJpZCI6MSwiZXBhIjoxNzEzMzUxMDY3fQ.WXD0-EVnIuYr9acZH0JTkvg0AeizZWm-OYSn-QPCSn0",
  "token_type": "bearer"
}
```

Response fields

- **message:** A simple confirmation message.
- **access_token:** A JSON Web Token access token, to be used for user authentication.
- **token_type:** The type of the access token, typically `bearer`.

Responses

- 200 - Successful Response
 - **Response schema:** `application/json`
 - **message**
 - **Type:** String
 - **access_token**
 - **Type:** String
 - **token_type**
 - **Type:** String
- 422 - Validation Error
 - **Response schema:** `application/json`
 - **detail**
 - **detail.loc**
 - **Type:** Array of strings or integers
 - **detail.msg**
 - **Type:** String
 - **detail.type**
 - **Type:** String

[↑ Return to top](#)

Getting a User Profile

Request type: GET

API call

```
http://localhost:5000/api/v2/users/profile
```

Authorisations

- OAuth2PasswordBearer
 - **Flow type:** `password`
 - **Token URL:** `api/v2/users/login`

Example response

```
{
  "user": "string",
  "credits": 2000
}
```

Response fields

- **user:** The username of the account being returned.
- **credits:** The credit balanced of the account.

Responses

- 200 - Successful Response
 - **Response schema:** `application/json`
 - **credits**
 - **Type:** Integer
 - **username**
 - **Type:** String

[↑ Return to top](#)

Updating a User Profile

Request type: PUT

Request body schema: `application/json`

API call

```
http://localhost:5000/api/v2/users/profile
```

Authorisations

- OAuth2PasswordBearer
 - **Flow type:** `password`
 - **Token URL:** `api/v2/users/login`

Request body

- **username**
 - **Type:** String
 - **Required:** Yes
 - **Description:** A unique username.

Example response

```
{
  "user": {
    "username": "newusertest"
  },
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJzdHJpbmcyIiwiaWQiOiJEsImV4cCI6MTcxMzZMTA0NX0.FwLKZkLAB_grCzqHr9oO_B2PIH_xka3HfXSqsHCuRYA",
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJzdHJpbmcyIiwiaWQiOiJEsImV4cCI6MTcxMzZMTA0NX0.Kxvtp0nIY0vPp2GOQ-ZKkbALqQQGSEKo9JxvDz1
}
```

Response fields

- **user**
 - **user.username:** The new, updated username for the account.
- **access_token:** A JSON Web Token access token, to be used for user authentication.
- **refresh_token:** A JSON Web Token refresh token, to be used to acquire additional access tokens.

Responses

- 200 - Successful Response
 - **Response schema:** application/json
 - **user**
 - **user.username**
 - **Type:** String
 - **access_token**
 - **Type:** String
 - **refresh_token**
 - **Type:** String
- 422 - Validation Error
 - **Response schema:** application/json
 - **detail**
 - **detail.loc**
 - **Type:** Array of string or integers
 - **detail.msg**
 - **Type:** String
 - **detail.type**
 - **Type:** String

[↑ Return to top](#)

Deleting a User Profile

Request type: DELETE

API call

```
http://localhost:5000/api/v2/users/profile
```

Authorisations

- OAuth2PasswordBearer
 - **Flow type:** password
 - **Token URL:** api/v2/users/login

Example response

```
{
  "message": "User deleted successfully."
}
```

Response fields

- **message:** A simple confirmation message.

Responses

- 200 - Successful Response
 - **Response schema:** application/json
 - **message**
 - **Type:** String

[↑ Return to top](#)

Getting a User's Credit Balance

Request type: GET

API call

```
http://localhost:5000/api/v2/credits
```

Authorisations

- OAuth2PasswordBearer
 - **Flow type:** password
 - **Token URL:** api/v2/users/login

Example response

```
{
  "credits": 2000
}
```

}

Response fields

- `credits`: The account balance.

Responses

- 200 - Successful Response
 - Response schema: `application/json`
 - `credits`
 - Type: Integer

[↑ Return to top](#)

Purchasing User Credits

Request type: POST

API call

`http://localhost:5000/api/v2/credits/purchase?amount={amount}`

Authorisations

- OAuth2PasswordBearer
 - Flow type: `password`
 - Token URL: `api/v2/users/login`

Query parameters

- `amount`
 - **Type:** Integer
 - **Required:** Yes
 - **Description:** The amount of credits to purchase.

Example response

```
{
  "message": "100 credits purchased successfully."
}
```

Response fields

- `message`: A simple confirmation message.

Responses

- 200 - Successful Response
 - Response schema: `application/json`
 - `message`
 - Type: String
- 422 - Validation Error
 - Response schema: `application/json`
 - `detail`
 - `detail.loc`
 - Type: Array of string or integers
 - `detail.msg`
 - Type: String
 - `detail.type`
 - Type: String

[↑ Return to top](#)

Getting a User's Locations

Request type: GET

API call

`http://localhost:5000/api/v2/users/locations`

Authorisations

- OAuth2PasswordBearer
 - Flow type: `password`
 - Token URL: `api/v2/users/login`

Query parameters

- `page`
 - **Type:** Integer
 - **Required:** No
 - **Description:** The page number to retrieve.
- `limit`
 - **Type:** Integer
 - **Required:** No
 - **Description:** The number of locations to retrieve per page.

Example response

```
{
  "locations": [
    {
      "temperature": 280.12,
      "city": "Retford",
      "latitude": 53.31880187988281,
      "longitude": -0.9824000000953674,
    }
  ]
}
```

```
      "user_id": 1,
      "timestamp": "2024-04-17T19:38:08.276103",
      "country": "United Kingdom",
      "id": 1,
      "description": "It's a bit chilly out there in Retford, United Kingdom! Make sure to bundle up with a warm jacket and maybe even a scarf to stay cozy in this cool wee"
    }
  ]
}

"pages": 1
}
```

Response fields

- A list of locations.
 - `locations.temperature`: The temperature of the location, in Kelvin.
 - `locations.city`: The name of the location.
 - `locations.latitude`: The latitude of the user's IP address.
 - `locations.user_id`: The ID of the user's account.
 - `locations.timestamp`: The timestamp of the call request.
 - `locations.country`: The name of the country.
 - `locations.id`: The ID of the location in the database.
 - `locations.description`: The AI-generated description message from OpenAI.
 - `locations.longitude`: The longitude of the user's IP address.
- `pages`: The number of pages of locations.

Responses

- 200 - Successful Response
 - Response schema: `application/json`
 - `locations`
 - `locations.temperature`
 - Type: Number
 - `locations.city`
 - Type: String
 - `locations.latitude`
 - Type: Number
 - `locations.longitude`
 - Type: Number
 - `locations.user_id`
 - Type: Integer
 - `locations.timestamp`
 - Type: String
 - `locations.country`
 - Type: String
 - `locations.id`
 - Type: Integer
 - `locations.description`
 - Type: String
 - `pages`
 - Type: Integer
 - 422 - Validation Error
 - Response schema: `application/json`
 - `detail`
 - `detail.loc`
 - Type: Array of string or integers
 - `detail.msg`
 - Type: String
 - `detail.type`
 - Type: String

[↑ Return to top](#)

Adding a New Location

Request type: POST

API call

`http://localhost:5000/api/v2/users/locations`

Authorisations

- OAuth2PasswordBearer
 - Flow type: `password`
 - Token URL: `api/v2/users/login`

Example response

```
{
  "message": "Location added successfully",
  "weather_info": "You may want to bring a waterproof jacket and some rain boots if you're heading out in Retford today. Stay dry and cozy!",
  "temperature": 282.14,
  "latitude": 53.31880187988281,
  "longitude": -0.982400000953674,
  "ip": "109.147.83.165"
}
```

Response fields

- `message`: A simple confirmation message.
- `weather_info`: The AI-generated description message from OpenAI.
- `temperature`: The temperature of the location, in Kelvin.
- `latitude`: The latitude of the user's IP address.
- `longitude`: The longitude of the user's IP address.
- `ip`: The IP address of the user.

Responses

- 200 - Successful Response
 - Response schema: `application/json`
 - `message`

- Type: String

[↑ Return to top](#)

Deleting a User's Location

Request type: DELETE

API call

http://localhost:5000/api/v2/users/locations/{location_id}

Authorisations

- OAuth2PasswordBearer
 - Flow type: password
 - Token URL: api/v2/users/login

Example response

```
{
  "message": "Location deleted successfully."
}
```

Response fields

- message: A simple confirmation message.

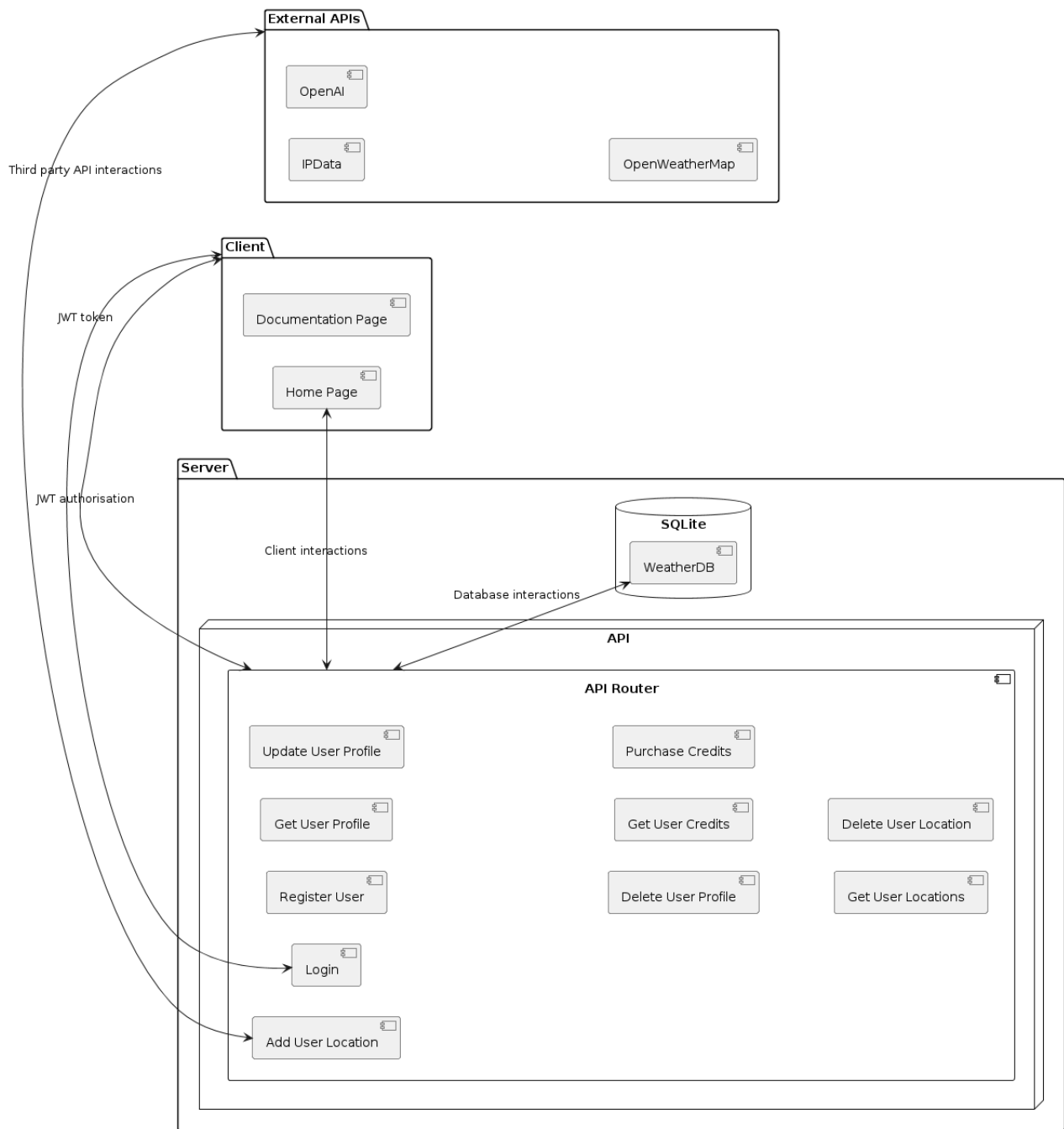
Responses

- 200 - Successful Response
 - Response schema: application/json
 - message
 - Type: String
- 422 - Validation Error
 - Response schema: application/json
 - detail
 - detail.loc
 - Type: Array of string or integers
 - detail.msg
 - Type: String
 - detail.type
 - Type: String

[↑ Return to top](#)

API Structure Diagram

Produced using UML and PlantUML.



[↑ Return to top](#)