

# Region and User Management

## API Documentation

Created By: Nafiz Imtiaz Khan

This API provides comprehensive functionality for managing geographical regions and user authentication. It features endpoints for retrieving weather forecasts, creating, reading, updating, and deleting region data, as well as user registration and login with JWT-based authorization.

This API facilitates the seamless integration of geographical region management, user authentication, and weather forecasting capabilities into external applications and systems. Its primary purpose is to provide a robust and secure interface for developers to access and manipulate region-specific data, manage user accounts, and retrieve relevant weather information. By offering endpoints for CRUD operations on region data, secure user authentication via JWT, and weather forecast retrieval, this API empowers developers to build location-aware applications, implement user access control, and enrich their services with real-time weather data. This integration enables diverse applications, ranging from location-based services and CRM systems to weather-focused mobile applications, to leverage a centralized and consistent data source for geographical regions, user management, and weather insights.

## Getting Started

### Endpoint & Environments

Base URL: <https://localhost:7142/>

### Authentication

This API uses JWT (JSON Web Token) authentication for secure access. Users must register and log in to obtain a token, which must be included in the headers of subsequent requests.

### Overview

The API provides endpoints for user registration and authentication. Upon successful login, the API returns a JWT token that must be used in the Authorization header for all authenticated requests.

Your token carries many privileges, so be sure to keep it secure! Do not share your token in publicly accessible areas such as GitHub, client-side code, and so forth.

This API allows new users to register by providing a username, password, and roles. Upon successful registration, the API returns a success message.

- **Endpoint:** **POST** {baseurl}/api/Auth/Register
- **Method:** **POST**
- **Content-Type:** application/json

**Request Body:**

Name	Type	Required	Description
username	string	Yes	The email or username of the new user
password	string	Yes	The user's password (must meet security requirements)
roles	array of strings	Yes	List of roles assigned to the user (e.g., admin, reader)

**Example Request Body (JSON):**

```
{
  "username": "karim@example.com",
  "password": "12345*",
  "roles": [
    "reader"
  ]
}
```

**Response:**

**Success Response (200 OK)**

"User Was Registered! Please LogIn"

**Get Weather Data:**

Overview

This API endpoint provides randomly generated weather data, including temperature values in Celsius and Fahrenheit, a brief weather summary, and the associated date.

**Endpoint:** [{baseurl}/weatherforecast](#)

**Method:** **GET**

**Response Details:**

**Success Response (200 OK)**

```
[{
  "date": "2025-02-24",
  "temperatureC": -11,
  "temperatureF": 13,
  "summary": "Hot"
},
{
  "date": "2025-02-25",
  "temperatureC": -4,
  "temperatureF": 25,
  "summary": "Cool"
}]
```

### Bad Request (**400**)

```
{
  "error": "Invalid request format",
}
```

### Not Found (**404**)

```
{
  "error": "Endpoint not found",
}
```

### Internal Server Error (**500**)

```
{
  "error": "An unexpected error occured",
}
```

### Response Fields

Field	Type	Description
date	string	The date for which weather data is generated (YYYY-MM-DD).
temperatureC	int	Temperature in Celsius.
temperatureF	int	Temperature in Fahrenheit.

## Get Region Data:

### Overview

This API retrieves **comprehensive region data**, structured as an array of objects. Each object contains:

- A **unique Region ID (GUID)**
- An **alphanumeric region code**
- A **descriptive region name**
- The **URL for the associated region image**

**Endpoint:** [{baseurl}/api/regions](#)

**Method:** **GET**

**Response Details:**

Success Response (200 OK)

```
[{
  "id": "0ada0027-8694-4cec-d890-08dd50d2769b",
  "code": "BDD",
  "name": "Bangladesh",
  "regionImgUrl": "bangladesh.jpeg"
},
{
  "id": "4220cf6c-e314-4e7e-d891-08dd50d2769b",
  "code": "BDS",
  "name": "badsbda",
  "regionImgUrl": "badasdasd"
}]
```

Bad Request (400)

```
{
  "error": "Invalid request format",
}
```

Not Found (404)

```
{
  "error": "Endpoint not found",
}
```

Not Found (500)

```
{
  "error": "An unexpected error occured",
}
```

**Response Fields**

Field	Type	Description
Id	Guid	Unique identifier for the region.
Code	String	Alphanumeric region code.
Name	String	Name of the region.
regionImgUrl	string	URL of the region's image.

## Create Region Data:

### Overview

This API allows you to create a new region by submitting a **POST** request with the required region details. The request must include:

- An **alphanumeric region code** (maximum 3 letters)
- A **descriptive region name**
- The **URL for the associated region image**

**Endpoint:** [{baseurl}/api/regions](#)

**Method:** **POST**

### Request Details:

Header: application/json

### Request Body (Json):

```
{
  "code": "SRL",
  "name": "Srilanka",
  "regionImgUrl": "srilanka.jpeg"
}
```

### Response Details:

Success Response (**200 OK**)

```
[{
  "id": "0ada0027-8694-4cec-d890-08dd50d2769b",
  "code": "BDD",
  "name": "Bangladesh",
  "regionImgUrl": "bangladesh.jpeg"
},
{
  "id": "4220cf6c-e314-4e7e-d891-08dd50d2769b",
  "code": "BDS",
  "name": "badsbda",
  "regionImgUrl": "badasdasd"
}]
```

### Bad Request (400)

```
{
  "type": "https://tools.ietf.org/html/rfc9110#section-15.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "errors": {
    "Code": [
      "Code Has To Be Maximum of 3 Letters"
    ]
  },
  "traceId": "00-5acf8c7f9fc0bd76f803fa4e1414b827-28c127a0bfb1374-00"
}
```

### Response Fields

Field	Type	Description
Id	Guid	Unique identifier for the region.
Code	String	Alphanumeric region code.
Name	String	Name of the region.
regionImgUrl	string	URL of the region's image.

### Get Region Data by Id:

#### Overview

This API retrieves detailed information about a specific region based on its unique identifier (**GUID**). The response includes the **region's ID, alphanumeric code, descriptive name, and associated image URL.**

**Endpoint:** `{baseurl}/api/regions/{id}`

**Method:** **GET**

#### Path Parameter:

Name	Type	Required	Description
id	string (GUID)	Yes	The unique identifier of the region

### Response Details:

#### Success Response (200 OK)

```
{
  "id": "0ada0027-8694-4cec-d890-08dd50d2769b",
  "code": "BDD",
  "name": "Bangladesh",
  "regionImgUrl": "bangladesh.jpeg"
}
```

Bad Request (**400**)

```
{
  "error": "Invalid request format",
}
```

Not Found (**404**)

```
{
  "error": "Region not found",
}
```

Internal Server Error (**500**)

```
{
  "error": "An unexpected error occured",
}
```

Response Headers

Header	Value
content-type	application/json; charset=utf-8
date	Sun, 23 Feb 2025 11:28:40 GMT
server	Kestrel

Update Region Data:

Overview

This API allows updating an existing region's details using a **PUT** request. The request requires a **region ID (GUID)** and includes updated values for **region code, name, and image URL**.

**Endpoint:** `{baseurl}/api/regions/{id}`

**Method:** **PUT**

Request Details:

Header: application/json

Path Parameter:

Name	Type	Required	Description
id	string (GUID)	Yes	The unique identifier of the region

**Request Body:**

```
{
  "code": "UAE",
  "name": "UnitedArabEmirates",
  "regionImgUrl": "uae.jpeg"
}
```

**Response Details:**

Success Response (200 OK)

```
{
  "id": "0ada0027-8694-4cec-d890-08dd50d2769b",
  "code": "UAE",
  "name": "UnitedArabEmirates",
  "regionImgUrl": "uae.jpeg"
}
```

Bad Request (400)

```
{
  "error": "Invalid request format",
}
```

Not Found (404)

```
{
  "error": "Region not found",
}
```

Unporcessable Entry (422)

```
{
  "error": "Validation failed: Code must be exactly 3 letters",
}
```

Internal Server Error (500)

```
{
  "error": "An unexpected error occured",
}
```

**Response Headers**

Header	Value
content-type	application/json; charset=utf-8
date	Sun, 23 Feb 2025 11:28:40 GMT
server	Kestrel



## Delete Region Data:

### Overview

This API allows deleting an existing region using a **DELETE** request. The request requires a **region ID (GUID)** as a path parameter. Upon successful deletion, the API returns the deleted region details.

**Endpoint:** `{baseurl}/api/regions/{id}`

**Method:** **DELETE**

### Path Parameter:

Name	Type	Required	Description
id	string (GUID)	Yes	The unique identifier of the region

### Response Details:

#### Success Response (**200 OK**)

```
{
  "id": "6884f7d7-ad1f-4101-8df3-7a6fa7387d81",
  "code": "NTL",
  "name": "Northland",
  "regionImgUrl": null
}
```

#### Bad Request (**400**)

```
{
  "error": "Invalid request format",
}
```

#### Not Found (**404**)

```
{
  "error": "Region not found",
}
```

#### Internal Server Error (**500**)

```
{
  "error": "An unexpected error occurred",
}
```

Response Headers

Header	Value
content-type	application/json; charset=utf-8
date	Sun, 23 Feb 2025 11:28:40 GMT
server	Kestrel

User Registration:

Overview

This API allows new users to register by providing a **username**, **password**, and **roles**. Upon successful registration, the API returns a success message.

Endpoint: {baseurl}/Auth/Register

Method: POST

Request Details:

Header: application/json

Request Body:

Name	Type	Required	Description
username	string	Yes	The email or username of the new user
password	string	Yes	The user's password (must meet security requirements)
roles	array of strings	Yes	List of roles assigned to the user (e.g., admin, reader)

Request Body (Json Format):

```
{
  "username": "karim@example.com",
  "password": "12345*",
  "roles": [
    "reader"
  ]
}
```

**Response Details:**

User Was Registered! Please LogIn

Success Response (200 OK)

```
{
  "username": "karim@example.com",
  "password": "12345*",
}
```

Bad Request (400)

```
{
  "error": "Invalid request format",
}
```

Conflict (409)

```
{
  "error": "User already exists",
}
```

Internal Server Error (500)

```
{
  "error": "An unexpected error occured",
}
```

**Response Headers**

Header	Value
content-type	application/json; charset=utf-8
date	Sun, 23 Feb 2025 11:28:40 GMT
server	Kestrel

**User Login:**

Overview

This API allows registered users to log in by providing their **username** and **password**. Upon successful authentication, a **JWT token** is returned, which can be used for authorization in subsequent requests.

**Endpoint:** {baseurl}/Auth/Login

**Method: POST**

### Request Details:

Header: application/json

### Request Body:

Name	Type	Required	Description
username	string	Yes	The email or username of the user
password	string	Yes	The user's password

### Request Body (Json Format):

```
{
  "username": "karim@example.com",
  "password": "12345*",
}
```

### Response Details:

User Was Registered! Please Login

## Success Response (200 OK)

```
{  
  "jwtToken":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzYyZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWVlc3ljbWVpbGFkZHJlc3MiOiJrYXJpbUBleGFtcGxlImNvbSIsImh0dHA6Ly9zY2hlbWFzLm1pY3Jvc29mdC5jb20vd3MvMjAwO  
C8wNi9pZGVudGl0eS9jbGFpbXMvem9sZSI6IlJlYWRlcisiImV4cCI6MTc0MDMxNDk2MywiaXNzIjoiaHR0cHM6Ly9sb2NhbgHvc3Q  
6NzE0Mi8iLCJhdWQiOiJodHRwczovL2xvY2FsaG9zdDo3MTQyLyJ9.h3AbiLOm6UrNPNGRG3_QsuCmvYRMLyrrSWy-n-BfeOrs"  
}
```

## Bad Request (400)

```
{
  "error": "Invalid request format",
}
```

## Unauthorized (401)

```
{
  "error": "Authentication failed",
}
```

Internal Server Error (500)

```
{
  "error": "An unexpected error occured",
}
```

Response Headers

Header	Value
content-type	application/json; charset=utf-8
date	Sun, 23 Feb 2025 11:28:40 GMT
server	Kestrel