API Exercise

In this exercise you will create APIs for an imaginary weather reporting module. The APIs load the given weather data in memory and return it for the given date range when requested. Use programming language of your choice to have working code, handling all error conditions that you can think of. Should follow best practices for REST API development. All APIs require authorization except the Token generation API. Use the correct choice of http method for each API.

Following APIs need to be created:

1. Generate Token - An API that takes username/password in the body and returns an access token in the response. Assume for now that if the username passed is **kirang**, an access token will be returned successfully, otherwise an error. The access token needs to be passed as a HTTP header in subsequent requests that require authorization

2. Save City Weather - This API takes the name of the city and corresponding weather data to save it in memory. The data is essentially a list of days and the mean temperature recorded on those days. If the city data already exists, it is updated.

3. Fetch City Weather - This API takes in the name of the city and a date range to return the temperatures recorded for those dates. To be able to fetch city weather, you need to save the city weather first (done by calling Save API). You should be able to handle error cases such as when city data has not been saved or when dates requested are out of range or invalid, using Bad request error. City names are case insensitive.

4. Delete City Weather - This API takes in the name of the city and deletes entire data corresponding to the city from memory. Fetching city weather after this operation results in a Bad request error. City names are case insensitive.

Assumptions:

1. Use JSON format for request and responses 2. Dates should be accepted in the format yyyy-MM-dd, ignore time zones for now. 3. Assume memory storage instead of database or file 4. Bad request should return HTTP 400 while unauthorized access returns HTTP 403 5. Calling an API without authorization returns an HTTP 403 error 6. Access token should be passed as a HTTP header in the request 7. Temperature can be assumed to be an integer 8. City names are case insensitive

Note: Please provide brief instruction on how to build your code and run it, if it isn't in a popular choice of programming language.

Weather Reporting API

Assumptions:

- Removing historic data for old days based on the request object
  Eg: Assume ‘Vancouver’ city has already reported for ‘April 17,2020’.If we have a new update request for ‘Vancouver’ does not contain entry for ‘April 17,2020’ then I am deleting the corresponding entry.
- Throwing error when there is a duplicate report for the same day of a city.
- 403 error is thrown by all API except Login API when access token of user other than “kirang” is passed
- Request and Response are using the same Model.

Requirements:

- Project is Build using ASP.NET Core 3.1
- Test Project is build using MS Tests
- Postman for API Testing (Note: **Turn OFF** SSL verification in Setting)
- Set Authorization= Bearer in Postman.

Endpoint	Method	Result	Note
/login	POST	Generates JWT token for user	Token is generated for user=”kirang”
/weatherreport	PUT	Save or update City Weather Report	
/weatherreport/city?startDate=yyyy-MM-dd&endDate=yyyy-MM-dd	GET	Gets City data for specified date range	Made Daterange optional.It gives full city weather report
/weatherreport	GET	Fetch weather report for all cities	
/weatherreport/city	POST	Deletes data report for specific city	

Sample Request:

```
        {
    "city": "seattle",
    "details": [
        {
            "date": "2020-04-17",
            "meanTemperature": 19
        },
        {
            "date": "2020-04-18",
            "meanTemperature": 18
        },
        {
            "date": "2020-04-19",
            "meanTemperature": 10
        },
        {
            "date": "2020-04-21",
            "meanTemperature": 10
        },
        {
            "date": "2020-04-28",
            "meanTemperature": 10
        }
    ]
}
```

Sample Response:

- Set Http Headers:
  [{"key":"Content-Type","value":"application/json"}]

- Authorization header is set for all API's except for login API
  [{"key":"Authorization","value": Bearer
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6ImtpcmFuZyIsImV4cCI6MTU4Nzc0OTQzNCwiaXNzIjoiVGVzdC5jb20iLCJhdWQiOiJUZXN0LmNvbSJ9.dUBH4yLOm-KLOp3f-syk5DIILTPUw-UOGN2e4Pkv84o"}]

Endpoint	Method	Request	Response
/login	POST	{ "UserName" : "kirang", "Password" : "password"	{ "token": "eyJhbGciOiJIUzI1NiIsInR

		}	5cCI6IkpXVCJ9.eyJ1bmlx dWVfbmFtZSI6ImtpcmFu ZyIsImV4cCI6MTU4Nzc0 OTQzNCwiaXNzIjoiVGVz dC5jb20iLCJhdWQiOiJUZ XN0LmNvbSJ9.dUBH4yL Om-KLOp3f-syk5DIILTPU w-UOGN2e4Pkv84o" }

/weatherreport	PUT	{
		"city": "seattle",
		"details": [
		{
		"date": "2020-04-17",
		
		"meanTemperature": 19
		},
		{
		"date":
		"2020-04-18",
		
		"meanTemperature": 18
		},
		{
		"date":
		"2020-04-19",
		
		"meanTemperature": 10
		},
		{
		"date":
		"2020-04-21",
		
		"meanTemperature": 10
		},
		{
		"date":
		"2020-04-28",
		
		"meanTemperature": 10
		}
		]
			{
			"city": "seattle",
			"details": [
			{
			"date":
			"2020-04-17T00:00:00",
			
			"meanTemperature": 19
			},
			{
			"date":
			"2020-04-18T00:00:00",
			
			"meanTemperature": 18
			},
			{
			"date":
			"2020-04-19T00:00:00",
			
			"meanTemperature": 10
			},
			{
			"date":
			"2020-04-21T00:00:00",
			
			"meanTemperature": 10
			},
			{
			"date":
			"2020-04-28T00:00:00",
			
			"meanTemperature": 10
			}

		}	] }
/weatherreport/vancouver?startDate=2020-04-17&endDate=2020-04-19	GET	Gets City dataspecified date range	{ "city": "vancouver", "details": [{ "date": "2020-04-17T00:00:00", "meanTemperature": 19 }, { "date": "2020-04-18T00:00:00", "meanTemperature": 18 }, { "date": "2020-04-19T00:00:00", "meanTemperature": 10 }] }
/weatherreport	GET	Fetch weather report for all cities	[{ "city": "vancouver", "details": [{ "date": "2020-04-17T00:00:00", "meanTemperature": 19 }, { "date": "2020-04-18T00:00:00", "meanTemperature": 18 },

			{ "date": "2020-04-19T00:00:00", "meanTemperature": 10 }, { "date": "2020-04-21T00:00:00", "meanTemperature": 10 }, { "date": "2020-04-28T00:00:00", "meanTemperature": 10 }] }, { "city": "seattle", "details": [{ "date": "2020-04-17T00:00:00", "meanTemperature": 19 }, { "date": "2020-04-18T00:00:00", "meanTemperature": 18 }, { "date": "2020-04-19T00:00:00", "meanTemperature": 10 }, { "date": "2020-04-21T00:00:00",

			"meanTemperature": 10
			},
			{
			"date":
			"2020-04-28T00:00:00",
			
			"meanTemperature": 10
			}
			]
			}
			]

/weatherreport/seattle	POST		seattle city weather report deleted