City Temperature Application Documents

This Application is used to get the temperature of a city. I have implemented a Restful web service using Java Api's(Jax-rs) and other public Apis to get the weather data(**OpenWeatherMap**) and capital of the country(**RestCountries.eu**).This web service can be called from any browser or tool to get the data.

Requirement Gathering :

simple API with 2 endpoints that returns JSON.

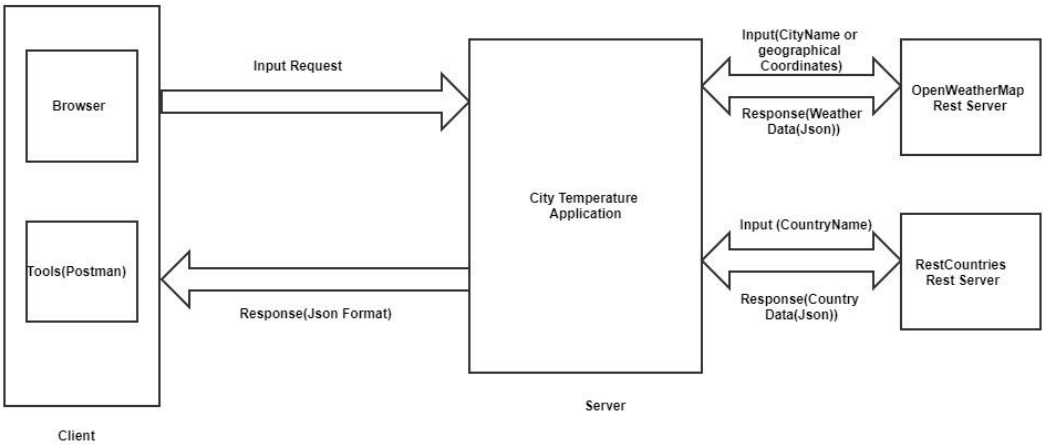
1. City Name with other geographic information(Latitude and Longitude) and User Local Time
2. Country Name (to determine Capital City Name) and User Local Time

Requirement Analysis :-

1. Developing Rest API to get the current temperature of the city using city name and country name
2. Determine the capital city of the country when user provide input country name
3. Find the difference between user local time and time of the city provided as input

Designing:

Designing consists of how different parts are interconnected to each other to communicate.
How request and response are exchanged between client and server.
Following diagram shows how the city temperature application Architecture..



Development:-

In the development phase I have implemented an actual application which will work as per the requirement mentioned.Following are the main technology and tools are used ti

Programming language:Java 11 JDK,Java Api(Jax-RS)
Build Tool:Maven
IDE:Eclipse
Application server:WildFly

Dependencies Used In Projects are as follows.

Following are the dependencies are used

1. **Javax**- To develop Rest Services
2. **Com.squareup.okhttp3 -** HTTP client to call public web service to get the weather and capital city data
3. **Org.json** - To Parse the response from public api's and then convert it into the customise json format response
4. **Log4j-** To log the logs of the application

Pubic Or OpenSource API's Used References:

To get the data of the weather of the city and to determine the capital of the country ,the following APIs are used.

1. **OpenWeatherMap**-To get the current temperature of the city(https://api.openweathermap.org/data/2.5/weather)
2. **RestCountries.eu :-** To get the capital city of the country(restcountries.eu)
   We have called this web service from our application only one per city.as =user search for a particular country,it will be cached into the application, so if the user searches for the same country again then we should not call the same web service for that country.

Rest EndPoints of the Application

Following are the Rest endpoint implemented:

1. **http://{hostname}:{port}/city-temperature/cityweatherapp/rest/v0/temperature/gettempbycityname**
2. **http://{hostname}:{port}/city-temperature/cityweatherapp/rest/v0/temperature/gettempbycountryname**

EndPoint1:-Get city Temperature using city name or geographical locations and get difference between user local time zone and city's local time

GET
http://{hostname}:{port}/city-temperature/cityweatherapp/rest/v0/temperature/gettempbycityname

In the EndPoint 1 consist of:-

1. Protocol used:-HTTP/HTTPS
2. Hostname- On which you server is deployed
3. Port-On which server is running
4. city-temperature- this is name of the war which we are going to run
5. Relative Path URL- URL defined for the endpoint to access it

Following are the input (Query Params)required to execute the service

Param Name	**Required**	**Meaning**
cityName	True(If geographical locations not provided) False(If geographical locations are provided)	Name of the city to get the temperature
latitude	true(if cityName is not provided) false(if cityName is provided)	Geographical location of the city
longitude	True(if cityName is not provided) false(if cityName is provided)	Geographical location of the city
untis	False	Unit for the Temperature(standard, metric, and imperial units are available) default is -metric
UserLocalTime	True	User local time(Format is compulsory in the format(“2021-08-02 23:55”))

Headers Required :

Header Name	**Meaning**
Accept	application/json
Content-type	application/json

Note-Systems default time zone is considered as user local time zone.

Example1(cityname):-http://localhost:8080/city-temperature/cityweatherapp/rest/v0/temperature/gettempbycityname?cityName=Pune&unit=metric&userLocalTime=2021-08-02 23:55

Example2(geo coordinates)-http://localhost:8080/city-temperature/cityweatherapp/rest/v0/temperature/gettempbycityname?unit=metric&latitude=18.5196&longitude=73.8553&userLocalTime=2021-08-02 23:55

Note:Output for above both the examples will be the same.

Output:-


```
{
   "cityDetails": {
      "cityName": "Pune",
      "countryName": "IN",
      "geographicalCoordinates": {
         "latitude": 18.5196,
         "longitude": 73.8553,
         "timeZoneOffset": 19800
      },
      "temperature": 27.26
   },
   "differenceInCityTimeAndUserLocalTimeInHours": 37,
   "userDetails": {
      "userLocalTime": "2021-08-02 23:55",
      "userLocalTimeZoneOffset": 19800
   }
}
```


Output Details:-

City Details:-

Name -name of the city
Country name-name of the country in which city present
Geographical coordinates- latitude,longitude and timezone offset

userDetails:-

userLocalTime-local time of the user
userLocalTimeZoneOffset- zone offset of users local time

differenceInCityTimeAndUserLocalTimeInHours:-Difference between user local time and city's local time

EndPoint2:-Get capital city Temperature using country name and get difference between user local time zone and city's local time

GET
http://{hostname}:{port}/city-temperature/cityweatherapp/rest/v0/temperature/**gettempbycountryname**

In the EndPoint 1 consist of:-

6. Protocol used:-HTTP/HTTPS
7. Hostname- On which you server is deployed
8. Port-On which server is running
9. city-temperature- this is name of the war which we are going to run
10. Relative Path URL- URL defined for the endpoint to access it

Following are the input (Query Params)required to execute the service

Param Name	**Required**	**Meaning**
countryName	True	Name of the country to get the temperature of capital city
untis	False	Unit for the Temperature(standard, metric, and imperial units are available) default is -metric
UserLocalTime	True	User local time(Format is compulsory in the format("2021-08-02 23:55"))

Note-Systems default time zone is considered as user local time zone.

Headers Required :

Header Name	**Meaning**
Accept	application/json
Content-type	application/json

Example1(countryName):http://localhost:8080/city-temperature/cityweatherapp/rest/v0/temperature/gettempbycountryname?countryName=Ireland&userLocalTime=2021-08-03 12:55

Output:-


```
{
   "cityDetails": {
      "cityName": "Dublin",
      "countryName": "US",
      "geographicalCoordinates": {
         "latitude": 37.7021,
         "longitude": -121.9358,
         "timeZoneOffset": -25200
      },
      "temperature": 13.88
   },
   "differenceInCityTimeAndUserLocalTimeInHours": 12,
   "userDetails": {
      "userLocalTime": "2021-08-03 12:55",
      "userLocalTimeZoneOffset": 19800
   }
}
```


Output Details:-

City Details:-

Name -name of the capital city
Country name-name of the country in which city present
Geographical co-ordinates- latitude,longitude and timezone offset

userDetails:-

userLocalTime-local time of the user
userLocalTimeZoneOffset- zone offset of users local time

differenceInCityTimeAndUserLocalTimeInHours:-Difference between user local time and capital city's local time of the country

Configuration of Application:-

Application requires different configurations to execute.This configurations are stored in the **config.properties** in the application resource

Following configurations are stored in the config.properties.

1. **openweathermap.rest.api.key**=d6465f99e4ed2d2cc6fead1f648ac462
2. **openweathermap.rest.api.host.uri**=api.openweathermap.org
3. **openweathermap.rest.api.relative_url**=data/2.5/weather
4. **restcountries.rest.api.host**=restcountries.eu
5. **restcountries.rest.api.relative_url**=rest/v2/name

Note: These configurations are stored in the config file because, we can access this at multiple places and if in future we need to change this we have to change at one location only that is config.properties.
Also one you read any property from the file you don't have to again read the same property from the file as I have cached it in the Application.
This cache will definitely improve performance of the application.

Performance Improvement of Application:

To improve performance ,I have implemented a cache for the capital city of the country and read properties from the file.Following are the use of cache

- **Capital City Cache**:-As we know that the capital of the city is not going to change overnight,so I have cached them in the application corresponding to the country name. When a user search capital of the country first time it will be fetched from the Public rest Api i.e **restcountries.eu** and will be stored in the application cache,so whenever use tries to search same country, it will fetched from application cache only not from public rest api.
- **Properties Cache**:-Once any property from config.properties get accessed,it will stored in the cacheso, next time it will be fetched from the cache not from the config file.

Logging in the Application:

I have added log4j configuration for the application ,so we can have logs to debug the application whenever any issue comes up .
These logs can be used for analytics in the future.

How to build Executable War for Application:

To make it executable you need a maven library installed on your system.
Following are the steps to make it executable for the project.

1. Download Maven
2. Install Maven .check this Maven Installation
3. Clone Project from:-https://github.com/somnathmulay/CityTemperature
4. Go to the project directory. I.e project directory where you cloned the project .
5. Change **openweathermap.rest.api.key** property value to **d6465f99e4ed2d2cc6fead1f648ac462** in **src/main/resources/config.properties**
6. Run the command to create package(war) -mvn clean package

7. Step 6 will create target folder in same directory which will contains the **city-temperature.war** file

Deployment of Application Locally:

To deploy this application I used the **WildFly** web server.

Following are the steps to deploy applications locally using WildFly.

1. Download WildFly .check this-:WildFly
2. Extract Wildfly
3. Copy **city-temperature.war** from target folder of the project directory
4. Paste **city-temperature.war** into the **wildfly-24.0.1.Final/standalone/deployments** folder
5. Open cmd and move to the wildfly directory
6. Go to the **wildfly-24.0.1.Final/bin** of the extracted Wildfly server folder
7. Run standalone script as per you operating system i.e sh or bat
8. step 7 will start the server for the application on port 8080
9. Access wildfly server on http://localhost:8080
10. If you are able to access it that means your application is up and running
11. Now you can access the endpoints that is **cityName** and **CountryName** endpoints from postman or browser

Test Cases for the both end points:

1. Provide cityName as query param and user local time in the request
   -Correct output with all the details
2. Provide Geographical CoOrdinates i.e latitude and longitude and user local time-Correct output with the all details
3. Do not provide city name and geographical coordinates
   -response with error code and message
   Response:{
   "errorCode": "HTTP_400",
   "errorMessage": "Invalid input provided,Please provide at least city name(cityName) or geographical coordinates(latitude and longitude)"

   }

4.Provide cityName or geographical coordinates but do not provide the user local time

Response: {
"errorCode": "HTTP_400",
"errorMessage": "Invalid input provided,Please provide user local time as(userLocalTime)"
}

5.Provide cityName or geographical coordinates and user local time but in incorrect format

Response: {

"errorCode": "HTTP_400",

"errorMessage": "Invalid user local time format provided,Please provide user local time as(userLocalTime) in the format of 'yyyy-MM-dd HH:mm'"

}

6. Provide incorrect cityName or geographical coordinates and correct user local time

Response:{

"errorCode": "HTTP_404",

errorMessage": "City with the given name not found"

}

7.Provide correct countryName and userLocalTime

-Response with all details

8.Provide only userLocalTime

Response:

{

"errorCode": "HTTP_400",

"errorMessage": "Invalid input provided,Please provide country name as(countryName)"

}

9.Provide wrong country name and correct user local time

Response:

{

"errorCode": "HTTP_404",

"errorMessage": "Country with the given name not found"

}

Challenges:

1. First challenge was which public/open source rest api should be chosen to get the required data of the weather as per our inputs
2. Then how to get the capital city of the country using the country name.Which public/open source rest api will give the output we require
3. Determining the time zone of the city to get the local time
4. It was a little bit confusing what should be the input and output for each endpoint as it was dependent on the different scenarios how we think about the end points and also while selecting user local time as a user input or system date and time for user local time