**High Level Design for Weather Web Application**

Contents

1. Introduction 4

1.1 Document Purpose 4

2. Solution Design 5

3. Build Folder Structure 6

4. Appendix A - OpenWeatherMap API Output 7

4.1 Query weather by city name 7

4.2 JSON Weather Response 7

**History of Changes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Change** | **Author** |
| 1.0 | 05/06/2016 | Initial Document | Prasenjit Hazarika |

# Introduction

## Document Purpose

The purpose of this document is to define the solution design for Weather Web Application.

# Solution Design

The solution delivered under this design will leverage and obtained the weather data from online free weather API from [OpenWeatherMap](http://api.openweathermap.org/). The response and format of the data is in JSON.

The solution will be designed using Maven build framework and would be implemented using JSP based on HTML4/CSS to create the frontend UI and talk to online weather API. While it will invoke some backend java class methods to get the city list dynamically from a configurable city list.

The application will be delivery as .war package which will then have to be deployed by the customer to a Apache Tomcat Server.

below provides an illustration the instruction between different layers.

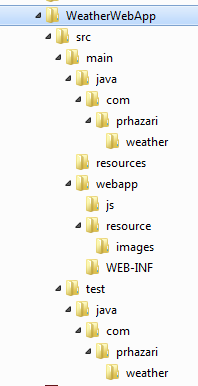


Figure 1: WeatherWebApp Overall Interaction Diagram

# Build Folder Structure

Below shows the project build folder structure for this solution:

It is based on Maven web application structure and will use maven scripts to build.



The production code will be stored inside ‘src/main’ while ‘src/test’ would have the unit test code.

The jsp and js files will reside under ‘src/main/webapp’ while java source files will be under ‘src/main/java’.

# Appendix A - OpenWeatherMap API Output

## Query weather by city name

Example query:

http://api.openweathermap.org/data/2.5/weather?"q=Sydney &APPID=73823783781&units=metric

## JSON Weather Response

{

"coord":{"lon":151.21,"lat":-33.87},

"weather":[{"id":521,"main":"Rain","description":"shower rain","icon":"09n"}],

"base":"stations","main":{"temp":14.95,"pressure":993,"humidity":93,"temp\_min":13.89,"temp\_max":16},

"visibility":10000,"wind":{"speed":10.3,"deg":270,"gust":15.4},"rain":{"1h":5.16},"clouds":{"all":75},

"dt":1465123424,

"sys":{"type":1,"id":8233,"message":0.0134,"country":"AU","sunrise":1465073646,"sunset":1465109596},

"id":2147714,"name":"Sydney","cod":200

}

Description of each field:

* coord
  + coord.lon City geo location, longitude
  + coord.lat City geo location, latitude
* weather (more info Weather condition codes)
  + weather.id Weather condition id
  + weather.main Group of weather parameters (Rain, Snow, Extreme etc.)
  + weather.description Weather condition within the group
  + weather.icon Weather icon id
* base Internal parameter
* main
  + main.temp Temperature. Unit Default: Kelvin, Metric: Celsius, Imperial: Fahrenheit.
  + main.pressure Atmospheric pressure (on the sea level, if there is no sea\_level or grnd\_level data), hPa
  + main.humidity Humidity, %
  + main.temp\_min Minimum temperature at the moment. This is deviation from current temp that is possible for large cities and megalopolises geographically expanded (use these parameter optionally). Unit Default: Kelvin, Metric: Celsius, Imperial: Fahrenheit.
  + main.temp\_max Maximum temperature at the moment. This is deviation from current temp that is possible for large cities and megalopolises geographically expanded (use these parameter optionally). Unit Default: Kelvin, Metric: Celsius, Imperial: Fahrenheit.
  + main.sea\_level Atmospheric pressure on the sea level, hPa
  + main.grnd\_level Atmospheric pressure on the ground level, hPa
* wind
  + wind.speed Wind speed. Unit Default: meter/sec, Metric: meter/sec, Imperial: miles/hour.
  + wind.deg Wind direction, degrees (meteorological)
* clouds
  + clouds.all Cloudiness, %
* rain
  + rain.3h Rain volume for the last 3 hours
* snow
  + snow.3h Snow volume for the last 3 hours
* dt Time of data calculation, unix, UTC
* sys
  + sys.type Internal parameter
  + sys.id Internal parameter
  + sys.message Internal parameter
  + sys.country Country code (GB, JP etc.)
  + sys.sunrise Sunrise time, unix, UTC
  + sys.sunset Sunset time, unix, UTC
* id City ID
* name City name
* cod Internal parameter