Overview

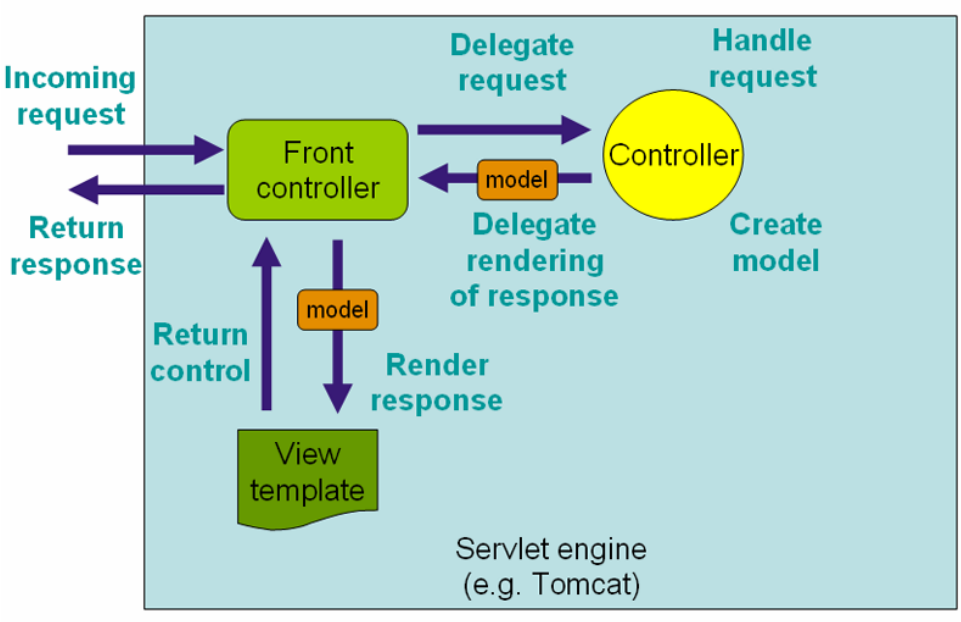
Weather Web App is a web application that tells you the current weather of the city your selected in the provided dropdown box in the Web User Interface.

Development Frameworks

This web application was developed using Spring, Spring MVC, Spring Boot, and Thymeleaf.

Spring Boot was used to deploy and run the application. Spring boot has an embedded web server like tomcat where this web application can be deployed.

Spring's web MVC framework is, like many other web MVC frameworks, request-driven, designed around a central Servlet that dispatches requests to controllers and offers other functionality that facilitates the development of web applications. Spring's DispatcherServlet however, does more than just that. It is completely integrated with the Spring IoC container and as such allows you to use every other feature that Spring has. (Pivotal)



The request processing workflow in Spring Web MVC (Pivotal)

Thymeleaf is a modern server-side Java template engine for both web and standalone environments.

Thymeleaf offers a set of Spring integrations that allow you to use it as a full-featured substitute for JSP in Spring MVC applications. (Thymeleaf)

This application used Thymeleaf to map methods in its Spring MVC **@Controller** objects forward to templates managed by Thymeleaf, exactly like you do with JSPs.

IDE

Eclipse with Spring Tool was used to develop this application.

Weather API

OpenWeatherMap weather API was used to get the current weather of a certain city.

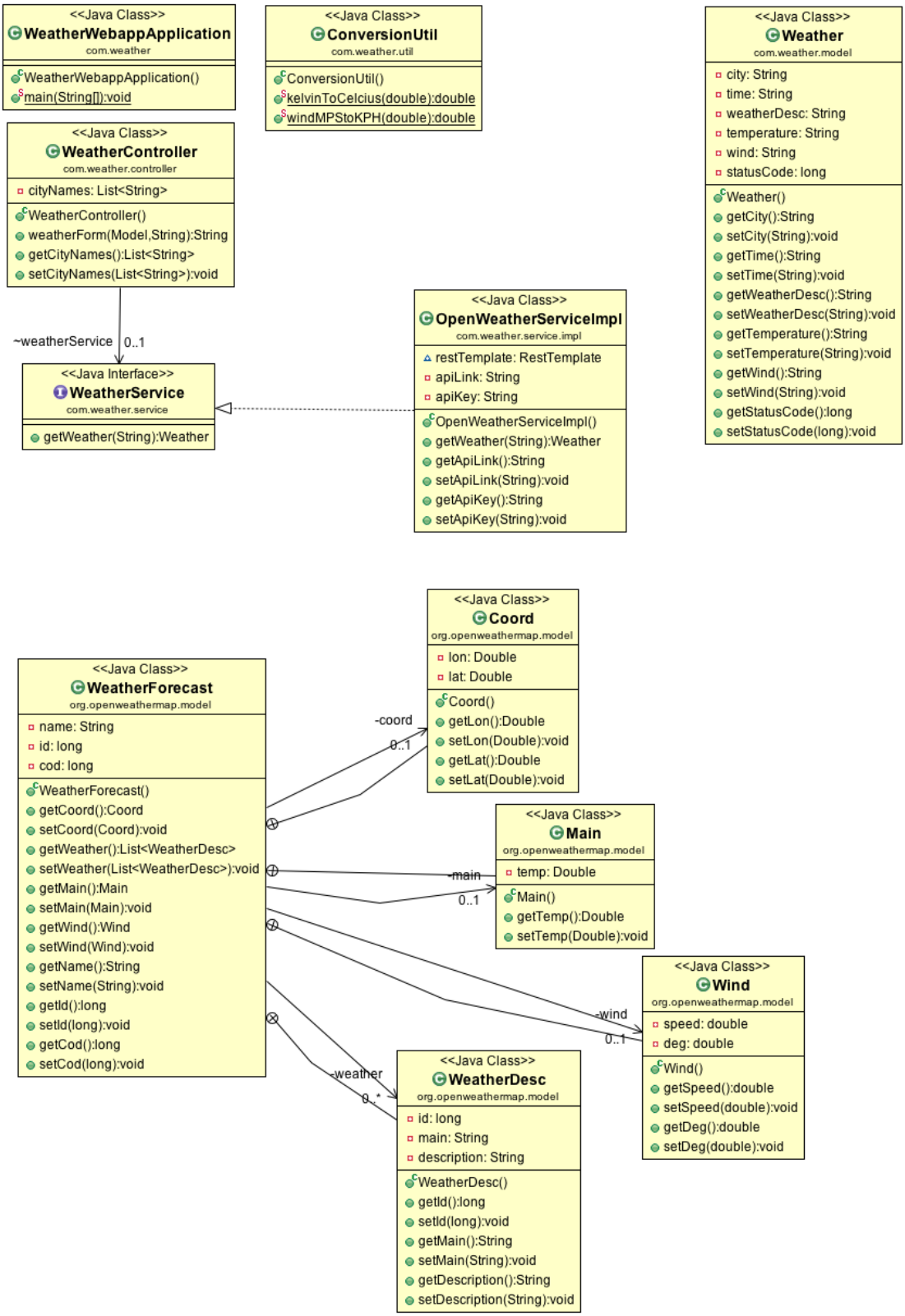
OpenWeatherMap is an open API for developers makes it easy to embed weather information into variety of applications including web and mobile weather applications, solutions for insurance, advertising, agriculture, sport and many others. For those who just want to know weather conditions for their location. Open Weather provides current weather, forecast, maps and a lot of charts on its web-site publicly. (Open Weather Map Inc.)

For further information please visit:
http://openweathermap.org/current

Assumptions

- Cities that will be added in the configuration file is always valid as the purpose of the dropdown box is to control input from user; therefore, the administrator of this application must always add valid cities.

Class Diagram



How to build the application

Use Maven to build the application.

Maven is a Java tool, so you must have Java installed in order to proceed.

1. First, download Maven and follow the installation instructions. After that, type the following in a terminal or in a command prompt:

```
mvn –version
```

It should print out your installed version of Maven, for example:

```
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T03:41:47+11:00)
Maven home: /Users/carizasy/Downloads/apache-maven-3.3.9
Java version: 1.8.0_91, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.10.5", arch: "x86_64", family: "mac"
```

2. Go to the application folder:

```
cd weather-webapp
```

3. Run the command:

```
mvn clean package install
```

4. Successful build must be like this:

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 16.602 s
[INFO] Finished at: 2016-05-XX T00:00:00+00:00
[INFO] Final Memory: 29M/269M
[INFO] ------------------------------------------------------------------------
```

Note: If build is unsuccessful, please contact cariza.sy@gmail.com

How to run the application

1. Go to the application target folder:

   ```
   cd weather-webapp/target
   ```

2. Run the command in your terminal or command prompt:

   ```
   java -jar demo-0.0.1-SNAPSHOT.jar
   ```

3. Spring Boot will start the application. The application is now accessible when you see the following lines at the bottom of your terminal or command prompt.

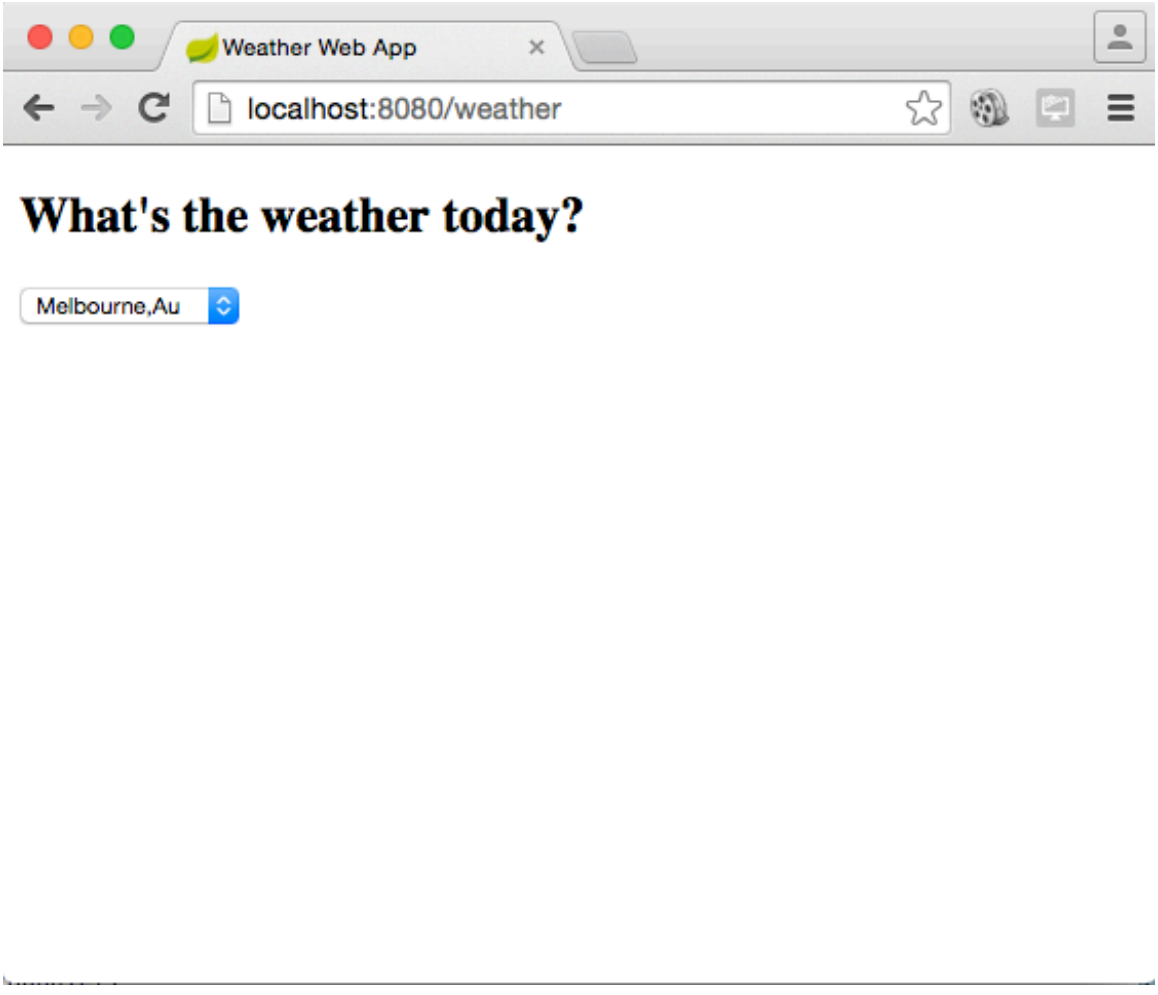
```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v1.3.3.RELEASE)


2016-05-02 15:27:05.365  INFO 7628 --- [           main]
o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path
[/**/favicon.ico] onto handler of type [class
org.springframework.web.servlet.resource.ResourceHttpRequestHandl
er]
2016-05-02 15:27:06.239  INFO 7628 --- [           main]
o.s.j.e.a.AnnotationMBeanExporter        : Registering beans for
JMX exposure on startup
2016-05-02 15:27:06.409  INFO 7628 --- [           main]
s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on
port(s): 8080 (http)
2016-05-02 15:27:06.421  INFO 7628 --- [           main]
com.weather.WeatherWebappApplication     : Started
WeatherWebappApplication in 8.467 seconds (JVM running for 9.349)
```

How to use the application

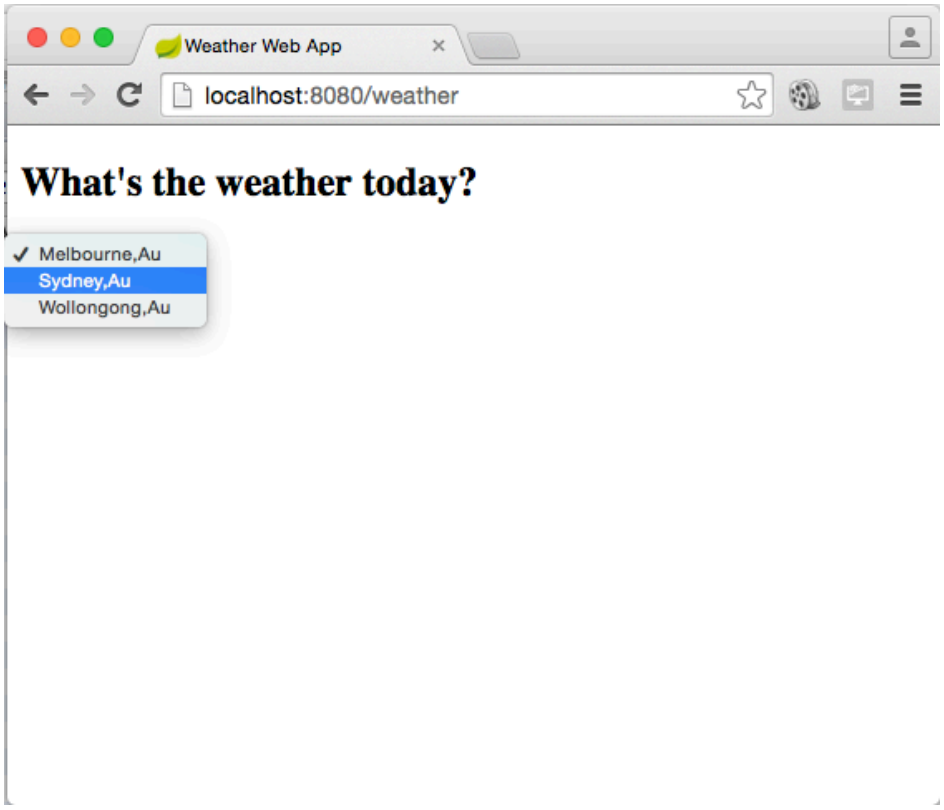
1. Open your web browser (e.g. Safari, Chrome)
2. Type in the following in your web browser's address bar:

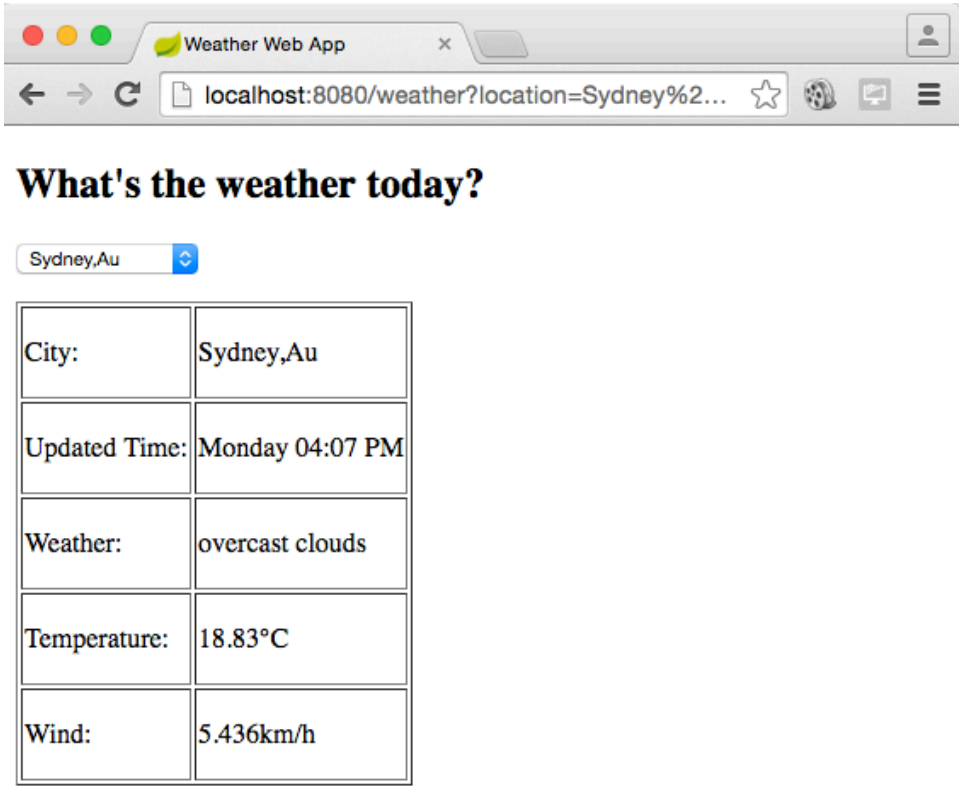
   http://localhost:8080/weather

3. After the UI has been rendered, it must look like this:

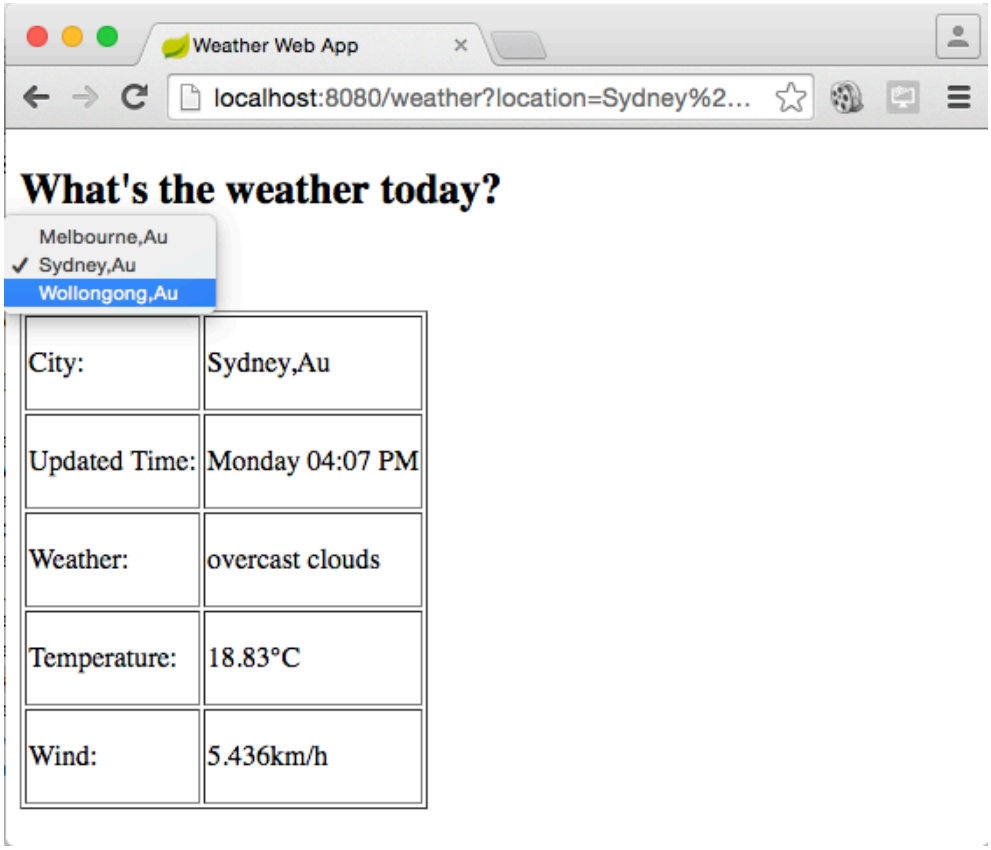


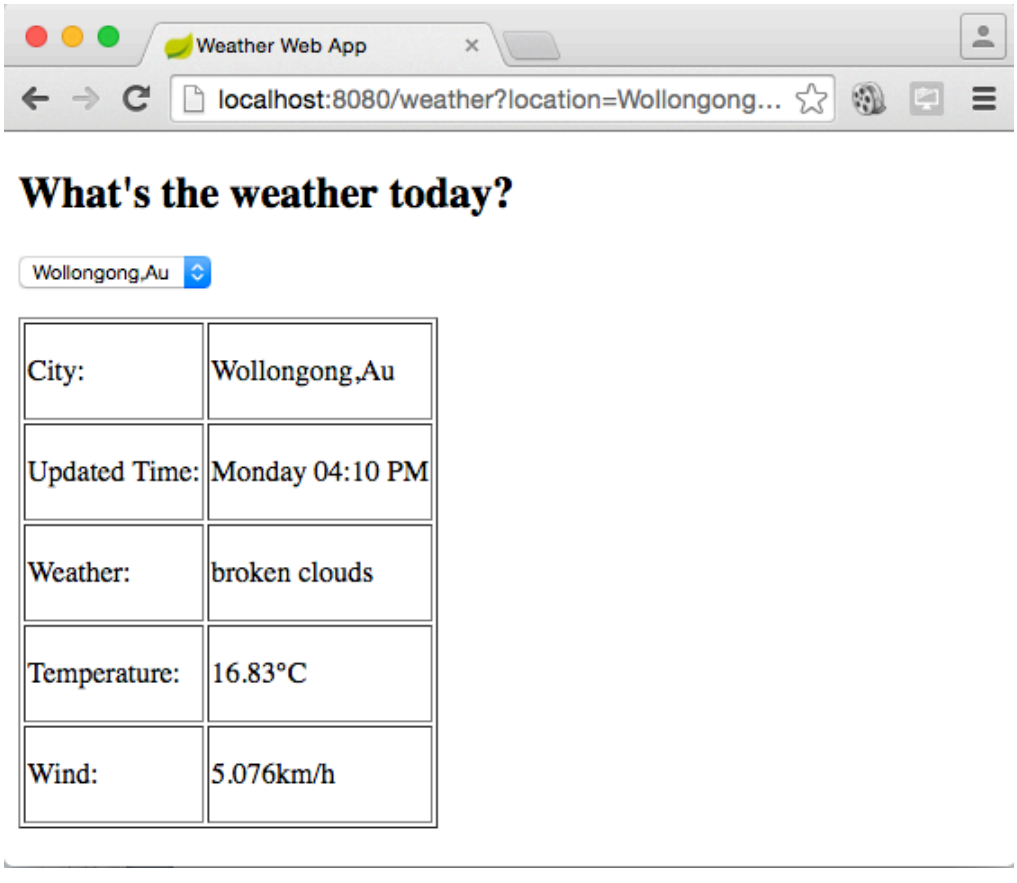
4. To get the weather details, click the dropdown box to select a city.





5. To get current weather to a different city again, just choose a different a city from the dropdown and the weather details will change after choosing.





How to add cities

1. Go to this folder:

   ```
   weather-webapp/src/main/resources
   ```

2. Edit the application.properties file.

   ```
   vi applications.properties
   ```

3. Add cities by using this format:

   ```
   ;city,countrycode
   ```

Example:
`weather.cityNames=Melbourne,Au;Sydney,Au;Wollongong,Au;Brisbane,Au`

4. Save changes

   ```
   :q!
   ```

5. Rebuild the application to apply changes.
6. Run the application.

Reference:

Pivotal. (n.d.). *Spring Web MVC Framework*. Retrieved 05 02, 2016, from http://docs.spring.io/autorepo/docs/spring/3.2.x/spring-framework-reference/html/mvc.html.

Thymeleaf. (n.d.). *Thymeleaf*. Retrieved 05 02, 2016, from http://www.thymeleaf.org.

Open Weather Map Inc. (n.d.). *Open Weather Map*. Retrieved 05 02, 2016, from http://openweathermap.org/technology.