Weather Application Design

Front-end Design

Libraries used for front-end development:

1. Angularjs 1.4.9

2. bootstrap 3.3.7

Front-end components

1. app.js

A global place for creating, registering and retrieving AngularJS modules. All modules (AngularJS core or 3rd party) that should be available to an application must be registered using this mechanism.

In this component there is factory which create a service to get the application base url and define wether service api base url.

2. weatherController.js

This component controls the data retrieved from the service layer, then format the data so that it can be displayed in required pattern. For example, time stamp will be formatted to the following pattern:

"DayofWeek hour:minute am/pm"

3. weatherService.js

The service component uses built-in service such as $http and $q to send HTTP GET request to the back-end and when response is received send the response to the controller layer.

4. cities.json

This file stores all the cities. If you want to add more cities and get their weather, just add city name to this file.

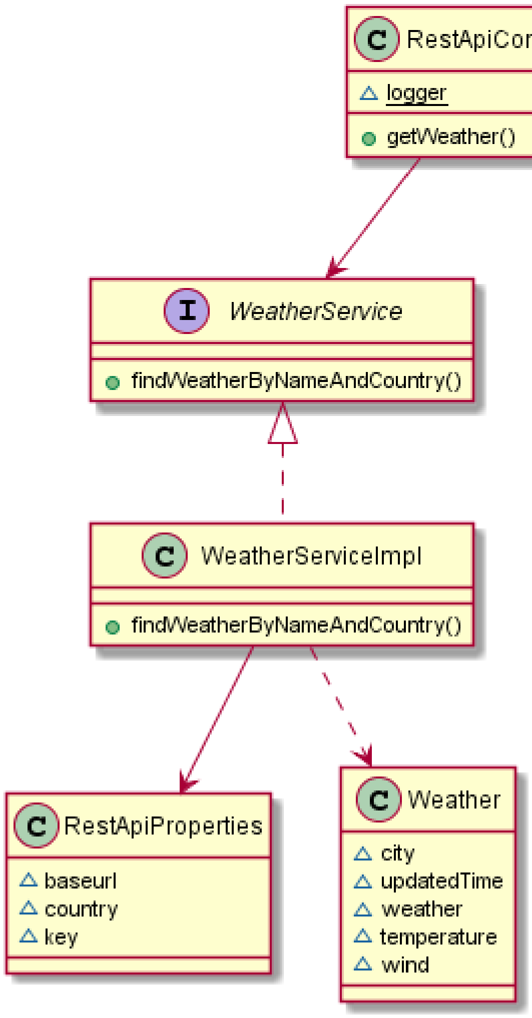
Back-end Design

Main Libraries used for back-end development:

1. Spring boot 1.4.3.RELEASE

2. Unirest for sending HTTP request in java

Class diagram



Back-end components

Back-end development follows MVC pattern, there are model, controllers and services.

1. RestApiController.java

This class is used for handling all rest api calls and call the service layer to retrieve data from the back end. ResponseEntity with data object from the service lay will be returned to the front-end.

2. WeatherService.java

The interface of the server layer. It has one method findWeatherByNameAndCountry(String city) which is used for getting the weather data.

3. WeatherServiceImpl.java

Service implementation class do to the real work. It will call Open Weather Map API to get the weather data.

4. Weather.java

Model class which used for representing the weather data information.

5. RestApiProperties.java

Data object used for representing application scale constants. The values in this object are read from the application.yml file.

6. CustomErrorType.java

A Custom class represents the error message.