**The Airport Weather Service API**

Table of Contents

[1. Overview 2](#_Toc470965546)

[2. REST API 4](#_Toc470965547)

[2.1 REST Airport Weather Collector 5](#_Toc470965548)

[2.1.1 Ping Alive 5](#_Toc470965549)

[2.1.2 Create Airport 5](#_Toc470965550)

[2.1.3 Get Airport 6](#_Toc470965551)

[2.1.4 Update Weather 7](#_Toc470965552)

[2.1.5 Delete Airport 8](#_Toc470965553)

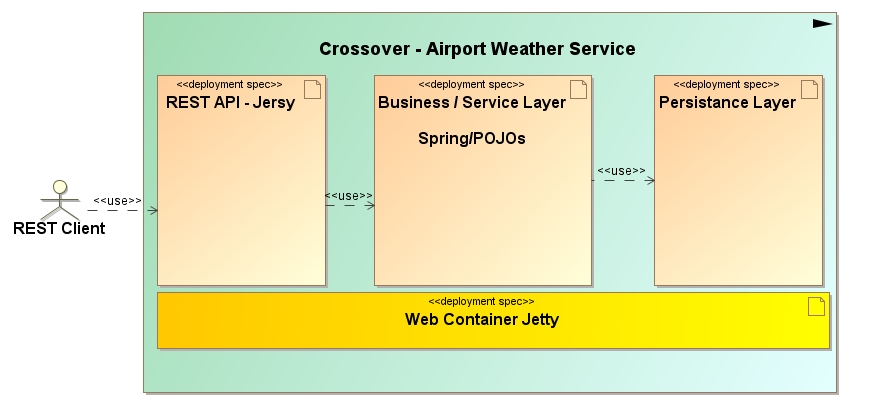
[2.2 REST Airport Weather Query 9](#_Toc470965554)

[2.2.1 Ping 9](#_Toc470965555)

[2.2.2 Query Weather 9](#_Toc470965556)

[2.3 Captured file 10](#_Toc470965557)

## Overview



The demo application uses a multi-layered architecture:

* the **first layer** is the REST support implemented with Jersey, has the role of a [facade](http://en.wikipedia.org/wiki/Facade_pattern) and delegates the logic to the business layer
* the **business layer** is where the logic happens
* the **data access layer** is where the communcation with the pesistence storage (in our case the map java database) takes place

The Airport Weather Service (AWS) is a REST application for collecting and redistributing meteorological data for a handful of airports. The service provides two distinct interfaces. One is a query interface, used by dozens of client systems to retrieve information such mean temperature and max wind speed. The other is a collector interface used by airports to update meteorological data stored in AWS.

**An airport is identified by an unique IATA code, longitude and latitude.**

\* **IATA code** : a three-letter code designating many airports around the world, defined by the International Air Transport Association (IATA).

\* **Longitude**: Decimal degrees, usually to six significant digits. Negative is West, positive is East.

\* **Latitude**: Decimal degrees, usually to six significant digits. Negative is South, positive is North.

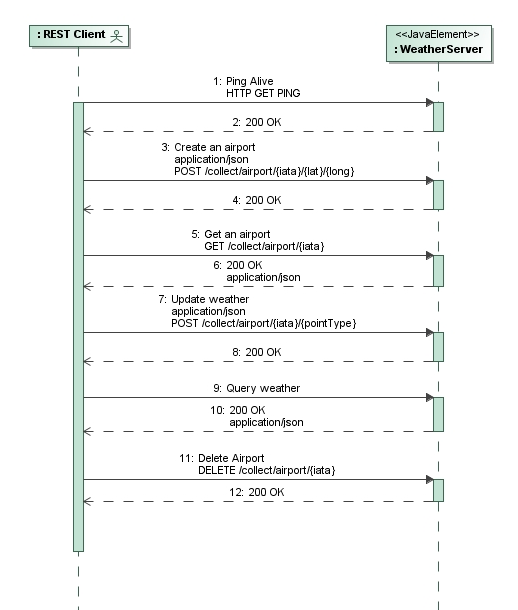
AWS is storing weather information of all airports in the AirportData. The information of each weather is mapping with the airport by iata code. There are six factors that determine the state and condition of the atmosphere and, therefore, influence and determine the weather. They include as following:  
\* 1. temperature  
\* 2. wind  
\* 3. humidity  
\* 4. precipitation  
\* 5. pressure  
\* 6. cloudCover

Each of the six factors is defined by a statistics atmospheric information as following:

* first 1st quartile -- useful as a lower bound.
* last 3rd quartile value -- less noisy upper value.
* mean the mean of the observations.
* median 2nd quartile -- median value.
* count the total number of measurements.

## REST API

REST client is communicating with the server using REST HTTP protocol. Consumers of the REST API can look at below diagram to understand API semantics. This client populates the REST endpoint with datatype specified by iata code.



## REST Airport Weather Collector

### Ping Alive

**GET /collect/ping**

A liveliness check for the collection endpoint.

|  |  |  |
| --- | --- | --- |
| **Parameters:** |  |  |
| **Status Codes:** |  | * [200 OK](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1) – the endpoint is alive. |

**Request**:

GET /collect/ping HTTP/1.1

User-Agent: Jersey/2.22 (HttpUrlConnection 1.8.0\_92)

Host: localhost:9090

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

**Response**:

HTTP/1.1 200 OK

Content-Length: 0

### Create Airport

**POST /collect/airport/{iata}/{lat}/{long}**

Add a new airport to the known airport list.

|  |  |  |
| --- | --- | --- |
| **Parameters:** |  | **iata** – the 3 letter airport code of the new airport  **lat** – the airport's latitude in degrees as a string [-90, 90]  **long** - the airport's longitude in degrees as a string [-180, 180] |
| **Status Codes:** |  | * [200 OK](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1) – added an airport successful. * [403](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.2) Forbidden – already exists the airport. * [406 Not](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5) Acceptable – invalid parameters: null or empty. |

**Request**: create an airport with parmeters "iataCode":"APK","latitude":49.0,"longitude":11.0

POST /collect/airport/**APK/49.0/11.0** HTTP/1.1

Content-Type: application/json

User-Agent: Jersey/2.22 (HttpUrlConnection 1.8.0\_92)

Host: localhost:9090

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

Content-Length: 51

{"iataCode":"APK","latitude":49.0,"longitude":11.0}

**Response**:

**HTTP**/1.1 200 OK

Content-Length: 0

### Get Airport

**GET /collect/airport/{iata}**

Add a new airport to the known airport list.

|  |  |  |
| --- | --- | --- |
| **Parameters:** |  | **iata** – the 3 letter airport code of the new airport |
| **Status Codes:** |  | * [200 OK](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1) – get an airport successful. * 404 Not found. * [406 Not](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5) Acceptable – invalid parameters: null or empty. |

**Request**:

GET /collect/airport/APK HTTP/1.1

User-Agent: Jersey/2.22 (HttpUrlConnection 1.8.0\_92)

Host: localhost:9090

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

**Response**:

HTTP/1.1 200 OK

Date: Fri, 30 Dec 2016 10:57:01 GMT

Content-Type: application/json

Content-Length: 47

{"iata":"APK","latitude":49.0,"longitude":11.0}

### Update Weather

**POST /collect/airport/{iata}/{pointType}**

Add a new airport to the known airport list.

|  |  |  |
| --- | --- | --- |
| **Parameters:** |  | **iata** – the 3 letter airport code of the new airport  **pointType** – a collected point |
| **Status Codes:** |  | * [200 OK](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1) – added an airport successful. * [403](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.2) Forbidden – already exists the airport. * [406 Not](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5) Acceptable – invalid parameters: null or empty. |

**Request**:

POST /collect/weather/APK/wind HTTP/1.1

Content-Type: application/json

User-Agent: Jersey/2.22 (HttpUrlConnection 1.8.0\_92)

Host: localhost:9090

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

Content-Length: 54

{"mean":6.0,"first":0,"median":4,"last":10,"count":20}

**Response**:

**HTTP**/1.1 200 OK

Content-Length: 0

### Delete Airport

**DELETE /collect/airport/{iata}**

Delete an airport with iata code.

|  |  |  |
| --- | --- | --- |
| **Parameters:** |  | **iata** – the 3 letter airport code of the delete airport |
| **Status Codes:** |  | * [200 OK](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1) – the endpoint is deleted. |

**Request**:

DELETE /collect/airport/MMU HTTP/1.1

User-Agent: Jersey/2.22 (HttpUrlConnection 1.8.0\_92)

Host: localhost:9090

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

**Response**:

HTTP/1.1 200 OK

Date: Fri, 30 Dec 2016 10:57:01 GMT

Content-Length: 0

## REST Airport Weather Query

### Ping

**GET /query/ping**

A liveliness check for the collection endpoint.

|  |  |  |
| --- | --- | --- |
| **Parameters:** |  |  |
| **Status Codes:** |  | * [200 OK](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1) – the endpoint is alive. |

**Request**:

GET /query/ping HTTP/1.1

User-Agent: Jersey/2.22 (HttpUrlConnection 1.8.0\_92)

Host: localhost:9090

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

**Response**:

**HTTP**/1.1 200 OK

Content-Length: 0

### Query Weather

**GET /query/weather/{iata}/{radius}**

Add a new airport to the known airport list.

|  |  |  |
| --- | --- | --- |
| **Parameters:** |  | **iata** – the 3 letter airport code of the new airport  **lat** – the airport's latitude in degrees as a string [-90, 90]  **long** - the airport's longitude in degrees as a string [-180, 180] |
| **Status Codes:** |  | * [200 OK](http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1) –successful. |

**Request**:

GET /query/weather/MMU/1000 HTTP/1.1

User-Agent: Jersey/2.22 (HttpUrlConnection 1.8.0\_92)

Host: localhost:9090

Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2

Connection: keep-alive

**Response**:

HTTP/1.1 200 OK

Date: Fri, 30 Dec 2016 10:57:01 GMT

Content-Type: application/json

Content-Length: 186

[{"temperature":null,"wind":{"mean":6.0,"first":0,"median":4,"last":10,"count":20},"humidity":null,"precipitation":null,"pressure":null,"cloudCover":null,"lastUpdateTime":1483095421552}]

## Captured file

A captured file by communicating between the client and the server.

It’d be opened by wireshark.

