28 Mar 2023

WEATHER

 VERSION: 1.1.0

# **Overview**

WeatherApp is a Foundry application that provides backend services to the users. The app contains a set of weather APIs. Using the APIs, users can obtain the weather information of any location in the world.

## **Use case:**

##  Consider a case that you want to develop an app that uses weather information. In this app, you want to develop a feature that helps users to plan daily activities based on the forecast of the weather conditions. Using the pre-configured services in the WeatherApp, you can obtain the required weather information

 You can import WeatherApp into your voltmx cloud account and access the services provided in WeatherApp. You can configure the services and achieve the desired results.

# **Getting Started**

## **Prerequisites**

## Before you start using the WeatherApp component, ensure the following:

* Volt MX Iris
* [VoltMX Foundry](https://manage.hclvoltmx.com/)

## **Platforms Supported**

### Mobile

#### iOS

#### Android

### Tablet

#### iOS

#### Android

## **Importing the app**

## The WeatherApp contains a set of integration services and orchestration services by default. These services fetch the required information using the Weather APIs. Using the Voltmx Foundry console, you can view and modify these services.

##  **To  import the Weather App Backend Service, do the following:**

1. Log on to your VoltMX Foundry. The **Dashboard** page appears by default.
2. In the left pane, click **Apps** menu . The **Custom Apps** page appears.
3. Click **IMPORT**. The **Import App** dialog appears.
4. Click Browse.
5. Select the **WeatherApp** zip file located in your system and click **Open**.
6. In the **Import App** dialog, click **IMPORT**.

**To associate your app with the WeatherApp, do the following:**

1. In Voltmx iris, open your app to which you have added the component.
2. On the **Projects** tab, click **Volt Foundry Backend**. The **Bind Volt Foundry Application** dialog appears.



1. Click **Use Existing**. The **Volt Foundry Applications** dialog appears with a list of existing apps.
2. Find the app, and click the **Associate** button against the app.

After you associate your app with the WeatherApp, you can view the integration services, and orchestration services, which are created by default. You can modify these services to fit your use case.

## **Integration Services**

The WeatherApp VoltMX Foundry app contains a set of integration services. The integration services are configured by default to connect your app to the Weather APIs. The integration service is responsible to fetch search results from the respective service provider.

To view the WeatherApp integration services, do the following:

1. Log on to your VoltMx account. the Dashboard page appears by default.
2. In the left pane, click the **Apps** menu. The **Custom Apps** page appears.
3. Find and click the **WeatherApp** app. The **WeatherApp** page appears along with the **Configure Services** tab and the **Identity** sub-tab opened by default.
4. Click the **Integration** sub-tab. The **Integration** tab displays a list of integration services.
5. Click on the required integration service. The **Service Definition** tab of the selected integration service opens by default.

The WeatherApp contains the following Integration Services:

#### AccuWeather

#### openWeatherMap

#### noaaWeather

#### worldweather

#### wunderground

#### Apple weather

### **AccuWeather**

### The accuWeather integration service fetches the weather information of any location in the world using the AccuWeather APIs. You need **Location Key** to use the integration service.

### The accuWeather integration service contains getForecast operation.

### The **getForecast** operation contains the following input parameters.

### **locationKey**: Obtains the location key. By default, the **VALUE** of the locationKey is set to **session**.

### **key**: Specifies the API key. To generate the key. <https://developer.accuweather.com/>

### The output of the operation depends on the service provider. For more information,refer <https://developer.accuweather.com/accuweather-forecast-api/apis>

The following procedure explains how to execute the **getForecast** operation and view the response:

1. Click the **Integration** sub-tab. The **Integration** tab displays a list of integration services.
2. Click **accuWeather**. The Service Definition tab of the selected integration service opens by default.
3. Click **Operations List** tab. The **Operations List** tab opens.
4. In the **Configured Operations** section, click **getForecast**. The **getForecast** tab opens with the **Request Input** sub-tab and **Body** section opened by default.
5. Click **Fetch Response**. The **Backend Response** dialog appears with the response.

**Sample Code:**

You can generate the sample code for an integration service from VoltMX Foundry. Then use that code in the mobile app to invoke the integration service instance.

//Code to invoke parent integration service should be present to use below code.

serviceName = "accuWeather";

integrationObj = VMXFoundry.getIntegrationService(serviceName);

operationName = "getForecast";

data= {"locationKey":"<Location Key>","key":"<API KEY>"};

headers= {};

integrationObj.invokeOperation(operationName, headers, data, operationSuccess, operationFailure);

function operationSuccess(res){

 //code for success call back

}

function operationFailure(res){

 //code for failure call back

}

1. **OpenWeatherMap**

The openWeatherMap integration service fetches the weather information of any location in the world using the OpenWeatherMap APIs.

The openWeatherMap integration service contains getForecast operation.

* The **getForecast** operation contains the following input parameters:
	+ **key**: Specifies the API key. To generate the key, visit <https://openweathermap.org/api>
	+ **lat**: Obtains the latitude of the location
	+ **long**: Obtains the longitude of the location
* The output of the operation depends on the service provider. For more information, refer <https://openweathermap.org/current>
* To execute the **getForecast** operation and view the response, follow the steps provided in the accuWeather integration service.

**Sample Code:**

* You can generate the sample code for an integration service from VoltMX Foundry. Then use that code in the mobile app to invoke the integration service instance.

//Code to invoke parent integration service should be present to use below code.
serviceName = "openWeatherMap";

integrationObj = VMXFoundry.getIntegrationService(serviceName);

operationName = "getForecast";

data= { key : <API KEY> ,"lat": "<Latitude>","long": "<Longitude>"};

headers= {};

integrationObj.invokeOperation(operationName, headers, data, operationSuccess, operationFailure);

function operationSuccess(res){

 //code for success call back

}

function operationFailure(res){

 //code for failure call back

}

1. **World weather**

The worldweather integration service fetches the weather information of any location in the world using the World Weather Online APIs.

The worldweather integration service contains getForecast operation.

* The **getForecast** operation contains the following input parameters:
	+ **Key** : Specifies the API key. To generate the key, visit <https://developer.worldweatheronline.com/signup.aspx>
	+ **lat** : Obtains the latitude of the location
	+ **long** : Obtains the longitude of the location
* The output of the operation depends on the service provider. For more information, refer <https://www.worldweatheronline.com/weather-api/api/docs/local-city-town-weather-api.aspx>
* To execute the **getForecast** operation and view the response, follow the steps provided in the accuWeather integration service.

**Sample Code:**

* You can generate the sample code for an integration service from VoltMX Foundry. Then use that code in the mobile app to invoke the integration service instance.

//Code to invoke parent integration service should be present to use below code.
serviceName = "worldweather";

integrationObj = VMXFoundry.getIntegrationService(serviceName);

operationName = "getForecast";

data= { key : <API KEY> ,"lat": "<Latitude>","long": "<Longitude>"};

headers= {};

integrationObj.invokeOperation(operationName, headers, data, operationSuccess, operationFailure);

function operationSuccess(res){

 //code for success call back

}

function operationFailure(res){

 //code for failure call back

}

###  **noaaWeather**

The noaaWeather integration service fetches the weather information of any location in the United States of America using [the NOAA National Weather Service APIs.](https://www.weather.gov/documentation/services-web-api#/)

The noaaweather integration service contains getForecast operation.

* The **getForecast** operation contains the following input parameters:
	+ gridId:
	+ gridX:
	+ gridY
* The output of the operation depends on the service provider. For more information, refer <https://www.weather.gov/documentation/services-web-api#/>

To execute the **getForecast** operation and view the response, follow the steps provided in the accuWeather integration service.

**Sample Code**:

You can generate the sample code for an integration service from VoltMX Foundry. Then use that code in the mobile app to invoke the integration service instance.

//Code to invoke parent integration service should be present to use below code.

serviceName = "noaaWeather";

integrationObj = VMXFoundry.getIntegrationService(serviceName);

operationName = "getForecast";

data= {"lat": "<Latitude>","long": "<Longitude>"};

headers= {};

integrationObj.invokeOperation(operationName, headers, data, operationSuccess, operationFailure);

function operationSuccess(res){

 //code for success call back

}

function operationFailure(res){

 //code for failure call back

}

1. **Wunderground**

The wunderground integration service fetches the weather information of any location in the world using the Wunderground APIs.

The wunderground integration service contains getForecast operation.

* The **getForecast** operation contains the following input parameters:
	+ **key**: Specifies the API key. To generate the key, visit <https://www.wunderground.com/weather/api>
	+ **lat**: Obtains the latitude of the location
	+ **long**: Obtains the longitude of the location
* The output of the operation depends on the service provider. For more information, refer <https://www.wunderground.com/weather/api/d/docs>

To execute the **getForecast** operation and view the response, follow the steps provided in the accuWeather integration service.

**Sample Code**:

You can generate the sample code for an integration service from VoltMX Foundry. Then use that code in the mobile app to invoke the integration service instance.

//Code to invoke parent integration service should be present to use below code.
serviceName = "wunderground";

integrationObj = VMXFoundry.getIntegrationService(serviceName);

operationName = "getForecast";

data= { key : <API KEY> ,"lat": "<Latitude>","long": "<Longitude>"};

headers= {};

integrationObj.invokeOperation(operationName, headers, data, operationSuccess, operationFailure);

function operationSuccess(res){

 //code for success call back

}

function operationFailure(res){

 //code for failure call back

}

1. **Apple weather**

Use the [WeatherKit REST API web service](https://developer.apple.com/documentation/weatherkitrestapi) to provide weather data to your apps and services that offer both current and forecasted weather information to your users. Create a [WeatherKit identifier and private key](https://developer.apple.com/documentation/weatherkitrestapi/request_authentication_for_weatherkit_rest_api) so you can use a developer token to authenticate yourself as a trusted developer and member of the Apple Developer Program. After getting identifier and private key generate web token.

* The **getForecast** operation contains the following input parameters:
	+ **lat**: Obtains the latitude of the location
	+ **long**: Obtains the longitude of the location
	+ add web token in the headers.



## **Orchestration Services**

Service orchestration is the coordination or integration of several services and exposing them as a single service. The mix of services supports the automation of business processes. For more information, refer [orchestration service.](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Foundry/vmf_integrationservice_admin_console_userguide/Content/Orchestration_Services.html)

The WeatherApp contains the following orchestration Service.

### **AccuWeatherService**

The AccuWeatherService fetches the weather information of any location in the world using the Accuweather APIs.

The service contains getForecast operation.

 The **getForecast** operation contains the following input parameters:

* + **key**: Specifies the API key. To generate the key, visit <http://developer.accuweather.com/>
	+ **lat**: Obtains the latitude of the location
	+ **long**: Obtains the longitude of the location

The output of the operation depends on the service provider. For more information, refer <https://developer.accuweather.com/accuweather-forecast-api/apis>

Sample Code:

You can generate the sample code for an orchestration service from Quantum Fabric. Then use that code in the mobile app to invoke the orchestration service instance.

//Code to invoke parent integration service should be present to use below code.

serviceName = "AccuWeatherService";

integrationObj =VMXFoundry.getCurrentInstance().getIntegrationService(serviceName);

operationName = "getForecast";

data= { key : <API KEY> ,"lat": "<Latitude>","long": "<Longitude>"};

headers= {};

integrationObj.invokeOperation(operationName, headers, data, operationSuccess, operationFailure);

function operationSuccess(res){

 //code for success call back

}

function operationFailure(res){

 //code for failure call back

}

# **Publishing the App in VOLTMX FOUNDRY**

After adding the **Weather App** Backend service to your app and configuring the necessary configurations, you must publish the app to VoltMX Foundry. For more information, refer to [Publish a foundry app.](https://opensource.hcltechsw.com/volt-mx-docs/docs/documentation/Foundry/voltmx_appfactory_user_guide/Content/FoundryPublish.html)

# **Revision History**

App version 1.1.0