Open Weather Map API Example

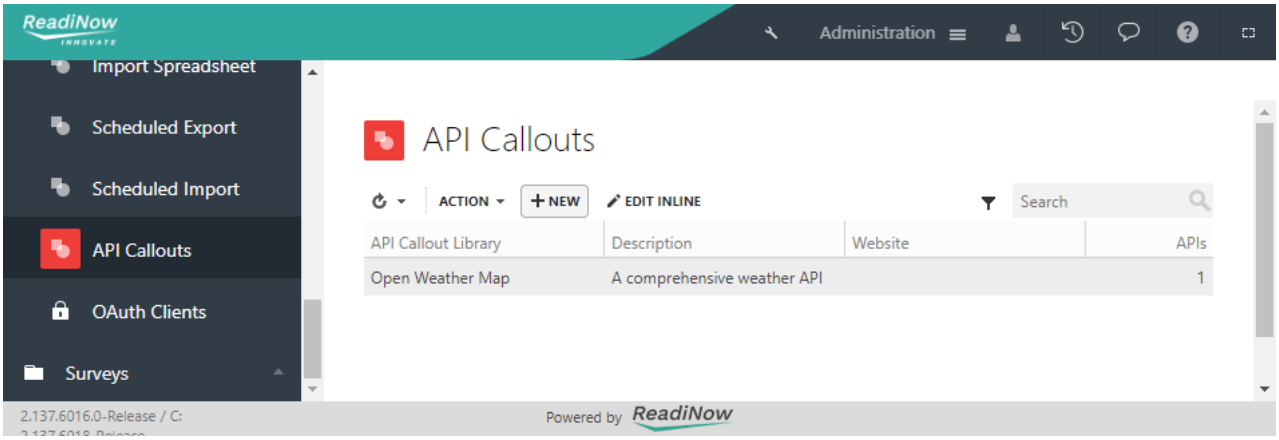
Last Modified on 23/08/2022 2:56 pm AEST

This article shows you how to setup a new API Callout Library and is based on the 'Open Weather Map' API. APIs solve the problem of how to connect an App to the internet - specifically 3rd party data source like: a weather forecast, sports-statistics, anything really.

The next sections outlines the basics of setting up an API (and provide links to more advanced information).

Create an 'API Callout Library' called "Open Weather Map"

When this step is completed the API Callouts form will look something like:

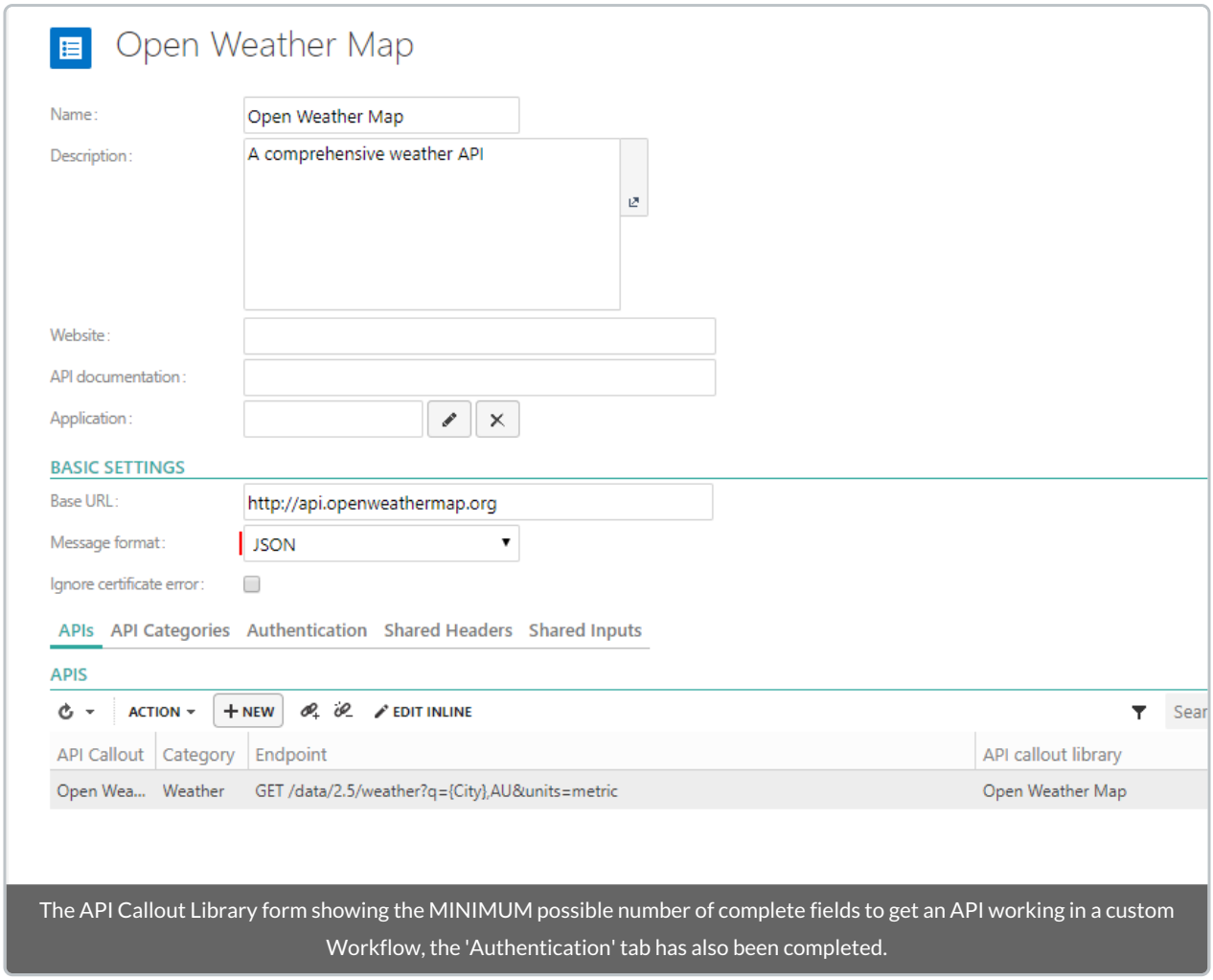


To achieve this:

- navigate to the API Callouts Page:

  Choose Application > Administration > Integration (left hand panel) > API Callouts
- Click on '+ New' (a blank Library form opens)

Once the minimum required information is completed, the completed 'Library Form' will look something like this:



The API Callout Library form showing the MINIMUM possible number of complete fields to get an API working in a custom Workflow, the 'Authentication' tab has also been completed.

The 'tabs' at the bottom section that we need to set up include: Authentication and APIs. These are covered in the remaining sections.

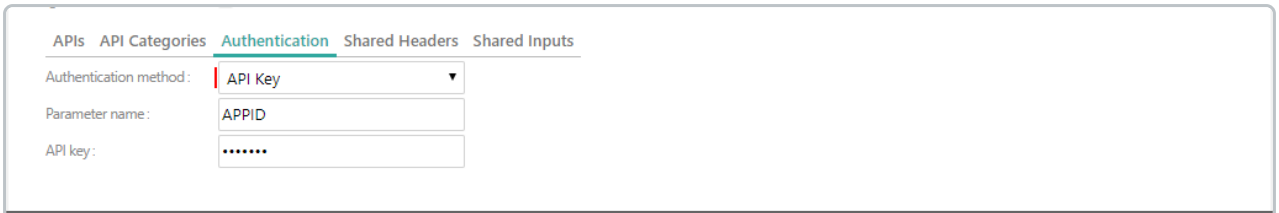
Learn about: advanced API configuration.

Setup the API-key *(provided by Open Weather Map)* in the API Callout Library

Learn about: other methods of authentication.

Before anything else we prefer to setup the API-key, this is how your Apps will Authenticate with the Open Weather Maps server.

In the image below, the 'parameter name' and 'API key' are provided by Open Weather Map; they also specify the authentication method.



The 'Authentication' tab is where you store your user credentials for the API, each of the API endpoints will use the same 'credentials'.

Add an '*API endpoint*' and '*sample response*' to the API Callout Library

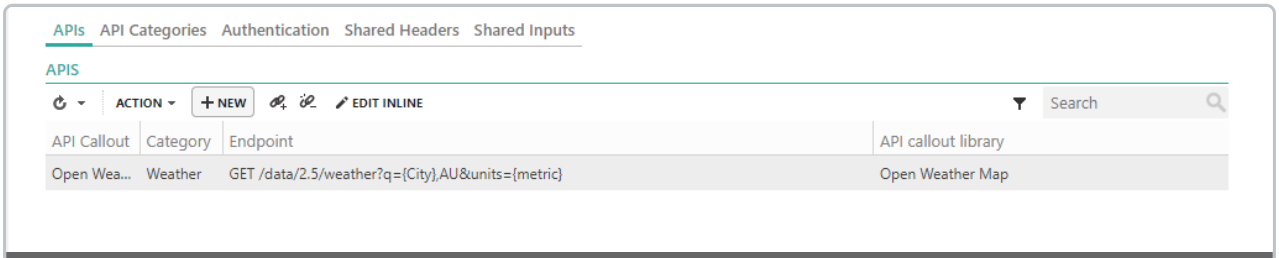
Learn about: using APIs.

API endpoint

An API endpoint is basically a website for Apps. The base address was set as a general library setting and all other URLs are relative to the base address. There are two points worth mentioning:

You can pass arguments to your endpoints; an argument is denoted by curly braces. For example if the endpoint could check the weather in any city in the world then you would create an argument like: {city}

The open tab 'APIs' shows our API Callout; it is named "Open Weather Maps", categorized as "Weather", and ends at a specified endpoint (of type 'GET').



The 'APIs' tab showing all the API endpoints in the selected 'Callouts Library' (in this case there is only 1 API endpoint). API Callouts are added by clicking on the '+New' button.

On the other hand, since our App is specially tweaked for 'Australian conditions' we do want

- to restrict the country code to "AU", and
- insist on using 'the metric system'

By the time the request is sent {city} will have a value, for example the user may have entered: "Nhulunbuy"; in that case, the API call looks like:

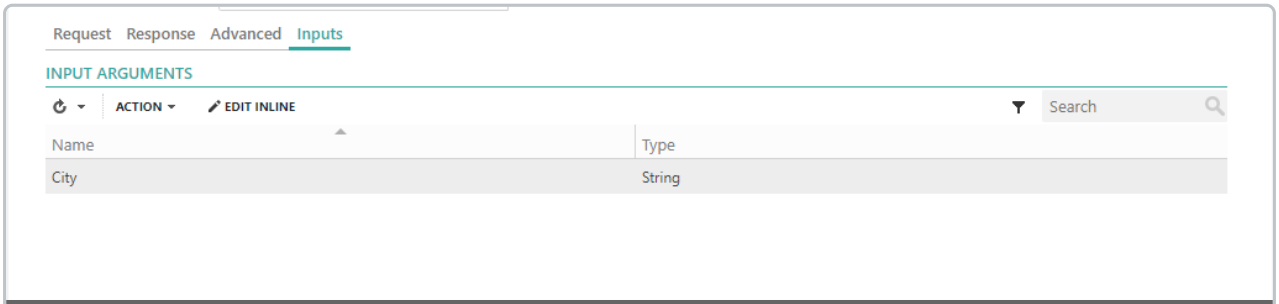
```
/data/2.5/weather?q=nhulunbuy,AU&units=metric
```

However, we don't always want to know the weather in Nhulunbuy so we can add a placeholder for the City Name {city}, this is shown below:

The API Callout form - open by double clicking on an existing API Callout OR create a new API Callout by pressing the '+New' button (on the parent form, not shown)

Configure Inputs

The inputs are created automatically when you use the curly braces { } when specifying the 'request url'. If they are not showing up press the refresh button.

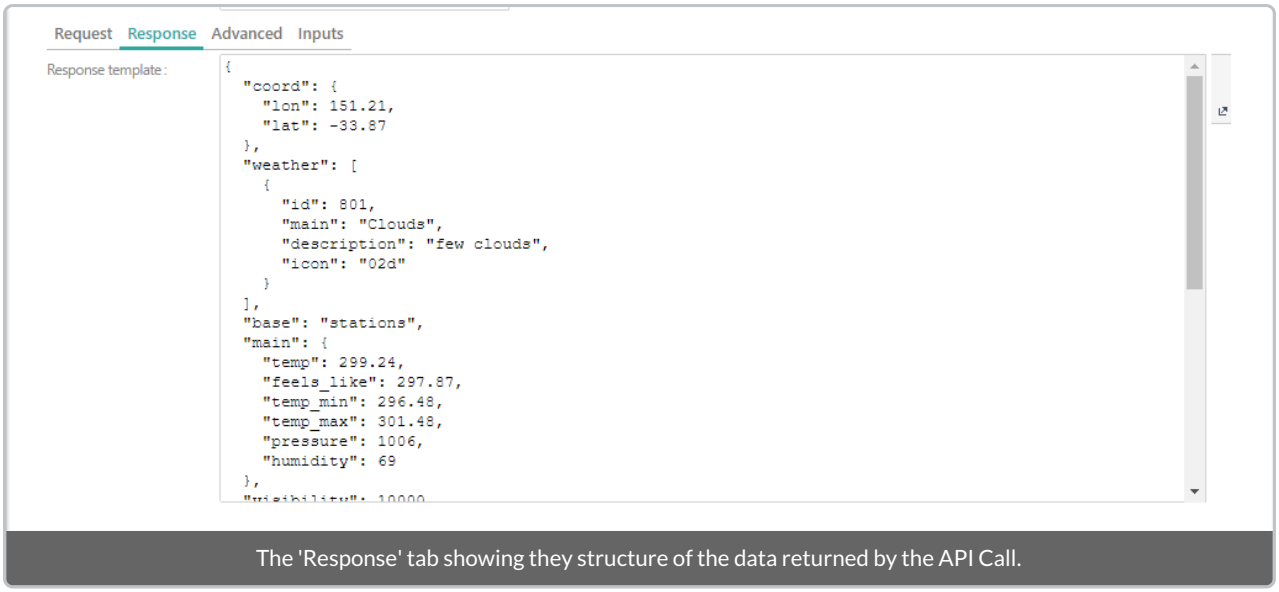


The 'Inputs' tab allows you input arguments to be assigned to a specific API Callout endpoint, the inputs were created automatically when "{city}" was typed into the 'Relative URL' in the previous image.
Note: Inputs ONLY apply to the selected API Callout - use 'Shared Inputs' for situations where Inputs apply to all APIs in the Callout Library.

Sample response

An API endpoint is basically a website for your App (except all the information is passed using a system of notation called JSON). In this example the 'Response template' is a JSON Object.

The sample response is literally *dummy-data*. What it does, however, is let you base decisions on the weather (or your own real-time analysis of your data) AND it helps with calculations.



The 'Response' tab showing they structure of the data returned by the API Call.