Local Skills - HelloWorldpart4_ExternalRequest

In part 4 of our *Hello World SDK* tutorial we'll learn how Misty can access external data from the internet and deliver it back to us. We'll write a skill that fetches the current temperature of a designated city, then sends it back through a debug message. To do this we need to use the `SendExternalRequest()` command, which we'll use to send a GET request to the APIXU API to get the data. We'll set the designated city as a default parameter in the meta file, as shown below.

HelloWorldpart4_ExternalRequest.json

```
{
      "Name": "HelloWorldpart4_ExternalRequest",
      "UniqueId": "523c7187-706e-4313-a657-0fa11d8bbdd4",
      "Description": "Local 'Hello, World!' tutorial series, part 4.",
      "StartupRules": [ "Manual", "Robot" ],
      "Language": "javascript",
      "BroadcastMode": "verbose",
      "TimeoutInSeconds": 300,
      "CleanupOnCancel": false,
      "WriteToLog": false,
      “Parameters”: {
            “city”: “Denver”
      }
}
```

HelloWorldpart4_ExternalRequest.js

The command `SendExternalRequest()` takes 6 required parameters (with two optional timing parameters at the end), which are outlined below:

```
misty.SendExternalRequest(method[string => POST,GET,DELETE,etc], Host URL
[string], "Resource URL [string]", "Authorization Type [string]", "token
[string]", "args [JSONstring]", "(optional) pre-pause[int]", "(optional)
post-pause[int]");
```

In this case we're sending a GET request, so we'll enter the string `GET` for the first parameter. The second parameter should contain the URL for the host of the API. In our case that is `http://api/apixu.com`.

The third parameter should contain the rest of the url required to hit the endpoint we're targeting. We'll designate the city we want to get weather data for using a template literal, passing the value `_params.city` into the URL string. The value `_params` gives us access to the `Parameters` value in the json meta file. We specified the key `city` to hold the city we want to search over. Here, `_params.city` holds the string `Denver.` The URL will also typically include an API key (which can easily be obtained through APIXU by registering for an account). For further information on constructing the query necessary for this activity, see APIXU's documentation here.

For some requests additional authorization may be necessary. This is where the forth and fifth arguments come in to play. However, in this case an API key is enough so we can enter `null` for those two parameters. The final parameter holds any data you are including in your request. In our case, this can also be set to `null` as we are sending a GET request. See below for reference:

```
misty.SendExternalRequest("GET", "http://api.apixu.com",
`/v1/current.json?key={{APIKEY}}&q=${_params.city}`, null, null, null);
```

Once Misty receives the data back the callback will run, which is automatically set to `_SendExternalRequest.`

```
function _SendExternalRequest(data) {}
```

At this point we have the data back from the send external request command. Like always, the response is an object. However, when Misty makes the external request, the data comes back in the form of a JSON string. So, within `data.Result` is a JSON string with the response from the API. Run `JSON.parse()` in order to convert it into an object and parse through to find the city name you searched with, and the current temperature of that city. Assign those to variables as shown below.

```
let currentCity = JSON.parse(data.Result).location.name;
let currentTemp = JSON.parse(data.Result).current.temp_f;
```

The last step is to have Misty send us the data back through a debug message:

```
misty.Debug(`The current temperature of ${currentCity} is ${currentTemp}
degrees fahrenheit.`);
```

In order to change the city Misty searches for, simply change the value of `city` within the parameters of your `meta` file. See the complete `.js` file below for reference.

```
// Debug message to indicate the skill has started
misty.Debug("starting skill helloworld part4");

// Send GET request to weather API
misty.SendExternalRequest("GET", "http://api.apixu.com",
`/v1/current.json?key=e1642c939804479ab3c213052182509&q=${_params.city}`, null,
null, null);

// Callback for external data
function _SendExternalRequest(data) {
   // Misty sends back the result from the command (SendExternalRequest) as an
object, but when Misty receives the data from the request, it is JSON string.
   // So, you need to parse the result from the request (data.Result) in order
to access any properties.

   // Assign variables to grab the city name and current temperature
   let currentCity = JSON.parse(data.Result).location.name;
   let currentTemp = JSON.parse(data.Result).current.temp_f;

    // Log message to display data to the user
   misty.Debug(`The current temperature of ${currentCity} is ${currentTemp}
degrees fahrenheit.`);
}
```