# Local 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 six required parameters (with two optional timing parameters at the end), which are outlined below:

```
misty.SendExternalRequest(method[string => POST,GET,DELETE,etc], "Resource URL [string]", "(optional)AuthorizationType [string]", "(optional)token [string]", "(optional)arguments[JSONstring]", "(optional)ReturnType [string]","(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 entire URI of the resource.

We can use variables to construct this URI. In the code, the `_params` object 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`. So, the URI of the resource is "`http://api/apixu.com/_params.city`".

Accessing an external resource may also require an API key (in this case, the key can easily be obtained through APIXU by registering for an account). Declare a global constant (`APIkey`) to hold your API key and pass it into the URL as shown below. For further information on constructing the query necessary for this activity, see APIXU's documentation here.

```
const APIkey = "<your API key here>";
```

Additional authorization may be necessary for certain requests. 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 arguments 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=${API key}&q=${_params.city}`, null, null, null, null);
```

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 to her in the form of a JSON string. So, `data.Result.Data` is a JSON string with the response from the API. Use `JSON.parse()` to convert the result into an object and parse through to find the city name and the current temperature of that city. Assign those to variables as shown below.

```
let result = JSON.parse(data.Result.Data);
let currentCity = result.location.name;
let currentTemp = 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");

const APIkey = "e1642c939804479ab3c213052182509";

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

// callback for external data
function _SendExternalRequest(data) {
    let result = JSON.parse(data.Result.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.Data) in
order to access any properties.

    // assign variables to grab the city name and current temperature
    let currentCity = result.location.name;
    let currentTemp = result.current.temp_f;

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