

HTTP Methods

Overview

Web services calls typically require some protocol to make requests. HTTP is an incredibly common protocol, so this page will introduce how to incorporate these calls in a python script. Note that all of the examples on this page can be easily called with the Script Console, but can be utilized through some other means (the actionPerformed event on a Button).

Finding an Endpoint

Ignition doesn't natively expose an endpoint for web services calls, so we'll have to utilize a service of some sort for the examples on this page. Fortunately, there are many public services we can utilize, such as [Yahoo Weather](#). From here we can generate an endpoint to use. At the time of this writing, we're using the following endpoint to retrieve the sunset time in Maui,HI:

Pseudocode - Finding a Resource

```
https://query.yahooapis.com/v1/public/yql?q=select%20astronomy.sunset%20from%20weather.forecast%20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text%3D%22maui%2C%20hi%22)&format=json&env=store%3A%2F%2Fdatatables.org%2Falltableswithkeys
```

On this page

...

- Overview
- Finding an Endpoint
- Making the Call
- Parsing the Results
- Make the Results Human Readable
- Troubleshooting HTTP Methods
 - HTTP Response Codes

Making the Call

To retrieve the results of this information in Ignition, we can use `system.net.httpGet()` to fetch the results of this call. We can try the following script in the [Scripting Console](#):

Python - Creates a variable to Store the Endpoint and Retrieves Results

```
# Create a variable to store the endpoint.
myEndpoint = "https://query.yahooapis.com/v1/public/yql?q=select%20astronomy.sunset%20from%20weather."
myEndpoint += "forecast%20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text%3D"
myEndpoint += "%22maui%2C%20hi%22)&format=json&env=store%3A%2F%2Fdatatables.org%2Falltableswithkeys"

print system.net.httpGet(myEndpoint)
```

Printing this results in the following Python string:

Python - Results

```
{"query":{"count":1,"created":"2017-10-31T16:44:19Z","lang":"en-US","results":{"channel":{"astronomy":{"sunset":"5:50 pm"}}}}}
```

Parsing the Results

If we wanted to extract a single value out of the results, we have a number of approaches. One useful approach would be to turn this JSON string into a Python Dictionary. This way we can single out a key instead of using regex or looking for substrings (both valid approaches in their own regard).

When presented with a JSON string, we can call `system.util.jsonDecode()` to turn a JSON string into a native Python Object. Thus, we can modify our code to the following:


```
Interactive Interpreter
Jython 2.5.3 (v2.5.3:3d2dbae23c52+, Nov 17 2012, 11:51:23)
[Java HotSpot(TM) 64-Bit Server VM (Oracle Corporation)] on java1.8.0_131

>>>
{  u'query': {  u'count': 1,
                u'created': '2017-10-31T16:44:19Z',
                u'lang': 'en-US',
                u'results': {  u'channel': {  u'astronomy': {  u'sunset': '5:50 pm'}}}}}}
None
>>> |
```

To get to the value for the sunset key, we simply need to address each key along the way:

```
Python - To Get Value for the Sunset Key
print decodedDict.get("query").get("results").get("channel").get("astronomy").get("sunset")
```

Troubleshooting HTTP Methods

When making HTTP calls, it is helpful to be familiar with the status codes that are returned by errors. To demonstrate, we could modify an earlier example:

```
Python - Returns Error Status Codes
# Create a variable to store the endpoint.
myEndpoint = "https://query.yahooapis.com/v1/public/yql?q=select%20astronomy.sunset%20from%20weather."

### Note that the following two lines are commented out, meaning we're about to make a GET call with a
bad/incomplete address.
# myEndpoint += "forecast%20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text%
3D"
# myEndpoint += "%22maui%2C%20hi%22)&format=json&env=store%3A%2F%2Fdatatables.org%2Falltableswithkeys"

print system.net.httpGet(myEndpoint)
```

This will return an error, which looks like the following:

```
Python - IOError: Response Code 400
Traceback (most recent call last):
  File "<buffer>", line 8, in <module>
IOError: Server returned HTTP response code: 400 for URL: https://query.yahooapis.com/v1/public/yql?
q=select%20astronomy.sunset%20from%20weather.
```

Note that HTTP response code **400**, which means bad request, was referenced. This error code is correct because we intentionally used an incomplete address!

HTTP Response Codes

The World Wide Web Consortium has a [page dedicated to HTTP response codes](#), which details all possible error codes. However, several common codes are listed below:

Response Code	Description
400	Bad Request - The server could not understand the request due to malformed syntax.

401	Unauthorized - The request requires some sort of authentication. Potentially some user credentials or an auth token of some kind.
403	Forbidden - The server understood what you requested, but is intentionally refusing the request. In some cases, the error message may include a reason why the request was not fulfilled (but not always). Typically, if the server doesn't include a reason, they'll use a 404 error code instead
404	Not Found - Your syntax was correct, but the server could not find the resource you were asking for. This could mean a typo, or missing portion of the URL you are using. In this case, double check the address you're specifying. Depending on the configuration, this could also mean that the server does actually have the resource you requested, but doesn't want to confirm its existence (typically due to a security policy. See error code 403 above).

Related Topics ...

- [system.net](#)