**Coding Test Project**

Congratulations!

Congratulations to making it onto our shortlist. As a potential candidate, we would like to give you the opportunity to show case your coding skills through a small practical project.

Overview

In brief, we would you to write us a small console-based application, which reads in a file containing details to one or more Restful API, creates a connection to these API(s) and checks the returned result is what we are expecting.

Basic Requirements

1. Read in a .json file (basic\_endpoints.json supplied).
2. For each “service”, initiate a Restful call to each endpoint listed for that service.
3. Evaluate the elements in the response received (JSON) from the endpoint to check if it is what we are expecting.

Additional Information

* The .json file contains a JSON array of services.
* Each service represents a single Restful API (though their couple be multiple endpoints).
* Each service has some basic elements as follows:

|  |  |
| --- | --- |
| baseURL | The base URL of the Restful API. All endpoints are found under this base URL. (i.e., https//mysite/api) |
| enabled | A flag to denoting if we should initiate calls to this service or skip it. |
| endpoints | A list of endpoints to check. |

* Each service could have one or more “endpoint”.
* Each endpoint has some basic elements as follows:

|  |  |
| --- | --- |
| resource | The resource representing this endpoint. (i.e., /account) |
| enabled | A flag to denoting if we should initiate calls to this endpoint or skip it. |
| response | A list of response elements we are expecting and how to evaluate them. |

* The full URL is thus made up of <baseURL>/<resource> (i.e., https//mysite/api/account).
* Each endpoint could have one or more “response”.
* Each response has some basic elements as follows:

|  |  |
| --- | --- |
| element | The response element we are evaluating. |
| regex | A regular expression used to evaluate the response. |

* Assume all requests are GET requests.

Example

|  |  |
| --- | --- |
| {  "services": [  {  "baseURL": "http://swapi.dev/api",  "enabled": true,  "endpoints": [  {  "enabled": true,  "resource": "/people/1",  "response": [  {  "element": "name",  "regex": "Luke Skywalker"  }  ]  }  ]  }  ]  } | * Make an HTTP GET request to http://swapi.dev/api/people/1 * Get the element “name” from the JSON response. * Check that the “name” supplied is “Luke Skywalker” |

Bonus Requirements

If you have your initiate implementation work well then, we strongly encourage you to try extending the project to include the following (see bonus\_endpoints.json):

* Add the ability to define a single set of “identifiers” per service, which can be re-used across the service endpoints when evaluating the responses.
* Add the ability to process both JSON and XML responses (add a “dataType” element to each service).
* … whatever you can think of that may be interesting / innovative to add.

Cheat Sheet

Reading the JSON file and initiating a request to the endpoint is the easy part, what we are really looking for is how you handle the response from the endpoints in a generic fashion (we not testing the against the same sample input).