Specs

**Intro**

When you create your request to the adapter, you supply all parameters in the body of the request, the same way the Bridge Adapter in a Chainlink does. Parameters are sent to the adapter in the “data” key of the body, and includes “spec” and “id1” […to “id6”], and an example could look like this:

{"data": {"spec": "firecontrol\_spec", "id1":"-120.2773332595825,50.202973148115944"}}

Specs

**Output Mapping**

Output Mapping is used to map the output of the individual APIs in the adapter into the final output that will be returned to the node job. Output Mappings are expressed using either a literal, a range of an API output or the “fullObj” key that when present, will make the adapter return a full object and not only a field.

**Literal:**

{"value":"0x"}

This will add the string “0x” to the output, in the selected sequence position

**Range:**

{"range":[0,64]}

This will add character positions 0-64 from the selected API into the selected position of the output

{"range":[0,10], "pad":"0", "padlen":32}

This will add first 10 characters from the selected API and pad with “0”es to a total length of 32

{"range":[0,0], "pad":"0", "padlen":16}

This will add all characters returned from the selected API and pad with “0” to a total length of 16

**Object:**

{"fullObj":true}

If the API returns a full object and not just a field, this spec config will return it to the final output. Can only be used with single-API adapters.

Specs

**Query Parameters**

Here we ask the adapter to use the spec called “firecontrol\_spec”, and we supply the id1 value of 120.277, 50.2029 which in this case represents a latitude/longitude. Query Parameters are used to tell the adapter how to map these parameters to the input parameters of the APIs in the adapter:

**Simple Mapping:**

{"c":{"id":"id1"}}

Here we tell the adapter that the c parameter for the API in question, should be taken from the id1 input to the adapter. This means that whatever we input to the adapter as id1 in the body, will be attached to the URL of the call to the API:

{“data”: { “spec”: “myspec”, “id1”: 12345} will be mapped to: myapi.com/getdata?**c=12345**

**Range Mapping:**

{"dt":{"range":[0,5],"id":"id1"}}

Here we tell the adapter that the dt parameter for the API in question, should be taken from the 0’th to the 5’th character of the id1 input to the adapter:

{“data”: { “spec”: “myspec”, “id1”: 987654} will be mapped to: myapi.com/getdata?**dt=98765**