

# How do I retrieve a value of a parameter within my API?

Web APIs can pass in parameters in three different ways: 1.) *Query Fields*, 2.) *Request Header Fields*, and 3.) in the *Request Body*. This article will explain **how to retrieve the value of a parameter** for each of these scenarios.

## Query Fields

Query fields (aka *query params*) appear in the URL and are separated from the main resource endpoint with the question mark ("?") character and then followed with one or more parameters. Each parameter is separated with the ampersand("&") character. Parameters normally take the form of a name /value pair. The equal sign("=") character is used to separate the field *name* from the field *value*. Query fields can appear alone as a label, but this rare and discouraged because the meaning of a label is ambiguous (e.g., *Does a label without a value imply an empty value? Is a label without a value meant to be a value in and of itself by virtue of appearing in the URL?*).

Here are a few examples of well-formed query fields:

```
GET https://www.examples.org/parts?color=red
```

```
GET https://www.examples.org/customers?status=active&state=NY
```

```
POST https://www.examples.org/oauth?token=5DS7o0QFQ2o1od2bI0KDKw9g1o3hMx
```

If query fields are important to your API (which is likely since [you need to allow them to be accepted in the first place](#)) you'll want to retrieve the value of each query field. This could be done by parsing the full URL yourself (which is automatically stored in the *FullEndpointURL* variable). However, we recommend you use the [GetQueryField](#) service. Here is what our code might look like:

```
// https://www.examples.org/parts?color=red
PartColor = HTTP_Services('GetQueryField', 'color')
CustStatus = HTTP_Services('GetQueryField', 'status')

// https://www.examples.org/customers?status=active&state=NY
CustState = HTTP_Services('GetQueryField', 'state')

// https://www.examples.org/oauth?token=5DS7o0QFQ2o1od2bI0KDKw9g1o3hMx
Token = HTTP_Services('GetQueryField', 'token')
```

 Query fields are case sensitive.

## Request Headers Fields

[HTTP requests](#) can also include [header fields](#). There are a number of [pre-defined header fields](#) but developers can also define their own.

 Please review the *Required Settings* section in the [Registry Configuration](#) article to properly enable custom header fields.

Here are a few examples of header fields:

```
Authorization: Basic dGVzdDp0ZXN0
Accept: image/jpeg
Accept-Encoding: gzip, deflate
```

Like query fields, header fields are also name/value pairs. However, these must be retrieved using the [GetRequestHeaderField](#) service:

```
Authorization = HTTP_Services('GetRequestHeaderField', 'Authorization')
Accept = HTTP_Services('GetRequestHeaderField', 'Accept')
AcceptEncoding = HTTP_Services('GetRequestHeaderField', 'Accept-Encoding')
```

 Unlike query fields, header fields are *not* case-sensitive.

## Request Body

While query fields and request header fields must conform to the patterns noted in the HTTP specification, parameters in the request body are always developer-defined and must be documented carefully. It is impossible to cover every possible response format and media type, so we'll focus on the two most likely cases: *JSON* (e.g., {"status" : "active"}) and *XML* (e.g., <status>active</status>). Regardless of the media type, use the [GetHTTPValue](#) service to retrieve the content of the body and then parse the data with the best tools available.

If the response body is formatted as JSON, we recommend using the [SRP\\_JSON](#) library to parse and retrieve the parameter value. Here is a simple example:

```
Body      = HTTP_Services('GetHTTPPostString')
If SRP_JSON(objBody, 'Parse', Body) EQ '' then
    CustStatus = SRP_JSON(objBody, 'GetValue', 'status')
    SRP_JSON(objBody, 'Release')
end
```

For an XML formatted response body, we recommend using the [SRP\\_Extract\\_XML](#) library. Again, here is a simple example:

```
Body      = HTTP_Services('GetHTTPPostString')
CustStatus = SRP_Extract_XML(Body, 'status/text()')
```