# API Jitterbit Variables

## Introduction

This page covers Jitterbit variables that are available for Jitterbit Custom APIs, organized by informational variables that you get (read), and settings variables that you set (write):

### Informational

- `jitterbit.api.request.body` and `jitterbit.api.request.body.*`
- `jitterbit.api.request.enum.body`
- `jitterbit.api.request.enum.headers`
- `jitterbit.api.request.enum.mvparameters`
- `jitterbit.api.request.enum.parameters`
- `jitterbit.api.request.headers.*`
- `jitterbit.api.request.headers.fulluri`
- `jitterbit.api.request.method`
- `jitterbit.api.request.mvparameters.*`
- `jitterbit.api.request.parameters.*`

### Settings

- `jitterbit.api.response`
- `jitterbit.api.response.headers.*`
- `jitterbit.api.response.status_code`

## Informational

### `jitterbit.api.request.body` and `jitterbit.api.request.body.*`

| | |
|---|---|
| **Variable** | `jitterbit.api.request.body` and `jitterbit.api.request.body.*` |
| **Data Type** | String |
| **Description** | Looks at the payload/payloads submitted to the API. Note that for the majority of the APIs, you would expect only one plain payload and, as such, `jitterbit.api.request.body` is the variable to use (also known as `content-type:text/plain`).<br><br>If you expect multiple payloads to be submitted at once, using the URL-encoded form (also known as `content-type:application/x-www-form-urlencoded`), as in the case of an API being used as the backend of a submission form (see http://www.w3.org/TR/html401/interact/forms.html), then you should be using `jitterbit.api.request.body.*`. As with `jitterbit.api.request.parameters.*`, `jitterbit.api.request.body.name` will be equal to `EStore` if the value of the form's field "name" was entered as `EStore`. |

### `jitterbit.api.request.enum.body`

| | |
|---|---|
| **Variable** | `jitterbit.api.request.enum.body` |
| **Data Type** | String |
| **Description** | Variable array used to dynamically iterate through all of the submitted parts of the payload/body (versus checking a specific part as with `jitterbit.api.request.body.*`). The usage is the same as with the `jitterbit.api.request.enum.parameters`. |

### `jitterbit.api.request.enum.headers`

| | |
|---|---|
| **Variable** | `jitterbit.api.request.enum.headers` |
| **Data Type** | String |

**Related Articles**

- API Jitterbit Variables
- Global Variable Source
- Global Variable Target
- Global Variable versus Temporary Storage
- Global Variables
- Hosted Endpoint Jitterbit Variables
- Jitterbit Variables
- Operation Jitterbit Variables
- Scripting Jitterbit Variables
- Source Jitterbit Variables
- Specifying Source and Target Fields Dynamically
- Target Jitterbit Variables
- Text Jitterbit Variables
- Transformation Jitterbit Variables
- Web Service Jitterbit Variables

**Related Topics**

- API Manager

Last updated: Feb 11, 2020

| | |
|---|---|
| **Description** | Variable array used to dynamically iterate through all of the request headers (versus checking specific headers as with `jitterbit.api.request.headers.*`). The usage is the same as with the `jitterbit.api.request.enum.parameters` and `jitterbit.api.request.enum.body`. |

### jitterbit.api.request.enum.mvparameters

| | |
|---|---|
| **Variable** | `jitterbit.api.request.enum.mvparameters` |
| **Data Type** | String |
| **Description** | Variable array used to dynamically iterate through all of the multi-value parameters (as opposed to checking each parameter specifically as `jitterbit.api.request.mvparameters.ProdID`). |

### jitterbit.api.request.enum.parameters

| | |
|---|---|
| **Variable** | `jitterbit.api.request.enum.parameters` |
| **Data Type** | String |
| **Description** | Variable array used to dynamically iterate through all of the submitted parameters (as opposed to checking each parameter specifically as `jitterbit.api.request.parameters.name`).<br><br>This sample script appends all of the provided parameters to a new variable for later display back to the user:<br><br>`<trans>`<br>`$output = "URL Parameters: <br>\r\n";`<br>`enum = $jitterbit.api.request.enum.parameters;`<br>`i = 0;`<br>`while(i<length(enum),`<br>`  name = enum[i];`<br>`  $output = $output + "$" + name + ": " + Get(name) + "`<br>`<br>\r\n";`<br>`  i = i+1;`<br>` );`<br>`if(i==0, $output = $output + "(none)<br>\r\n");`<br>`</trans>` |

### jitterbit.api.request.headers.*

| | |
|---|---|
| **Variable** | `jitterbit.api.request.headers.*` |
| **Data Type** | String |
| **Description** | Variable used to look at the request headers submitted to the API; for example, `$jitterbit.api.request.headers.x_forwarded_for` is the public IP of the box /user accessing the URL. |

### jitterbit.api.request.headers.fulluri

| | |
|---|---|
| **Variable** | `jitterbit.api.request.headers.fulluri` |
| **Data Type** | String |
| **Description** | The URL that was called to trigger the Jitterbit OData or Custom API. |

### jitterbit.api.request.method

| | |
|---|---|
| **Variable** | `jitterbit.api.request.method` |

| | |
|---|---|
| **Data Type** | String |
| **Description** | The request method that was used to call the API. |

### `jitterbit.api.request.mvparameters.*`

| | |
|---|---|
| **Variable** | `jitterbit.api.request.mvparameters.*` |
| **Data Type** | String |
| **Description** | Looks at the multi-values of the parameter submitted to the API directly via the URL and returns the values as an array with a space between each value.<br><br>For example, if the URL is `https://jitterbitxx.na.jitterbit.org/dev/ProductAPIResponse?ProdID=abc&ProdID=abc1&ProdID=abc2`, then `jitterbit.api.request.mvparameters.ProdID` will be `abc abc1 abc2`. |

### `jitterbit.api.request.parameters.*`

| | |
|---|---|
| **Variable** | `jitterbit.api.request.parameters.*` |
| **Data Type** | String |
| **Description** | Looks at the parameters submitted to the API directly via the URL; for example, `jitterbit.api.request.parameters.name` will be equal to `EStore` if the URL requested had `&name=EStore`.<br><br>**NOTE:** Multi-value URL parameters will return a string delimited by `\|\|\|` (3 pipes). To return multi-value URL parameters as an array, use the `jitterbit.api.request.mvparameters.*` variable instead.<br><br>For example, if the URL is `https://jitterbitxx.na.jitterbit.org/dev/ProductAPIResponse?ProdID=abc&ProdID=abc1&ProdID=abc2`, then `jitterbit.api.request.parameters.ProdID` will be `abc\|\|\|abc1\|\|\|abc2`. |

## Settings

### `jitterbit.api.response`

| | |
|---|---|
| **Variable** | `jitterbit.api.response` |
| **Data Type** | String |
| **Description** | This variable needs to be set if your API is set to use a variable as the response instead of using the final target. |

### `jitterbit.api.response.headers.*`

| | |
|---|---|
| **Variable** | `jitterbit.api.response.headers.*` |
| **Data Type** | String |
| **Description** | Used to set the response headers of the API. For example, set `jitterbit.api.response.headers.access_control_allow_origin="*"` to override default CORS behavior and allow the API to be accessed by any domain in a cross-site manner. |

### `jitterbit.api.response.status_code`

| | |
|---|---|
| **Variable** | `jitterbit.api.response.status_code` |
| **Data Type** | String |

| | |
|---|---|
| **Description** | Provides the ability to override HTTP response code for custom APIs via Jitterbit script variable. Set the `jitterbit.api.response.status_code` variable in the script that is executed by a Custom API. This allows project authors to set a specific HTTP error code (along with actual payload information) versus relying on the system to return codes 200 or 500 based on default behavior. |