

HTTP Header Prefix	Description
HTTP.REQ.HEADER("<header_name>")	Returns the contents of the HTTP header specified by the <header_name> argument. The header name cannot exceed 32 characters. Note that this prefix returns the value from the Host header by default. To use this value as a hostname you need to typecast it as follows: <code>http.req.header("host").typecast_http_hostname_t</code> .
HTTP.REQ.FULL_HEADER	Returns the contents of the complete set of HTTP header fields including the request line (for example, "GET /brochures/index.html HTTP/1.1") and the terminating <code>\r\n\r\n</code> sequence.
HTTP.REQ.DATE	Returns the contents of the HTTP Date header. The following date formats are recognized: RFC822. Sun, 06 Jan 1980 8:49:37 GMT, RFC850. Sunday, 06-Jan-80 9:49:37 GMT, ASCII TIME. Sun Jan 6 08:49:37 1980.
HTTP.REQ.COOKIE	(Name/Value List) Returns the contents of the HTTP Cookie header. HTTP.REQ.TXID Returns the HTTP transaction ID. The value is a function of an internal transaction number, system boot time and system MAC address
HTTP.RES.HEADER("<header_name>").	Returns the contents of the HTTP header specified by the <header_name> argument. The header name cannot exceed 32 characters
HTTP.RES.FULL_HEADER	Returns the contents of the complete set of HTTP header fields including the status line (for example, "HTTP/1.1 200 OK") and the terminating <code>\r\n\r\n</code> sequence.

<p>HTTP.RES.SET_COOKIE or HTTP.RES.SET_COOKIE2</p>	<p>Returns the HTTP Set-Cookie header object in a response.</p>
<p>HTTP.RES.SET_COOKIE(<name>) or HTTP.RES.SET_COOKIE2(<name>)</p>	<p>Returns the cookie of the specified name if it is present. If it is not present, returns a text object of length 0. Returns UNDEF if more than 15 Set-Cookie headers are present and the specified cookie was not found in these headers.</p>
<p>HTTP.RES.SET_COOKIE(<name>).DOMAIN or HTTP.RES.SET_COOKIE2(<name>).DOMAIN</p>	<p>Returns the value of the first Domain field in the cookie. For example, if the cookie is Set-Cookie : Customer = "ABC"; DOMAIN=".abc.com"; DOMAIN=.xyz.com, the following expression returns .abc.com: http.res.set_cookie.cookie("customer").domain. A string of zero length is returned if the Domain field or its value is absent.</p>
<p>HTTP.RES.SET_COOKIE.EXISTS(<name>) or HTTP.RES.SET_COOKIE2.EXISTS(<name>)</p>	<p>Returns a Boolean TRUE if a Cookie with the name specified in the `<name>` argument exists in the Set-Cookie header. This prefix returns UNDEF if more than 15 Set-Cookie headers are present and the named cookie is not in the first 15 headers.</p>
<p>HTTP.RES.SET_COOKIE.COOKIE(<name>).EXPIRES Or HTTP.RES.SET_COOKIE2.COOKIE(<name>)</p>	<p>Returns the Expires field of the cookie. This is a date string that can be evaluated as a number, as a time object, or as text. If multiple Expires fields are present, the first one is returned. If the Expires field is absent, a text object of length zero is returned.</p>
<p>HTTP.RES.SET_COOKIE.COOKIE(<name>).PATH PATH.GET(n)`.` HTTP.RES.SET_COOKIE2.COOKIE(\<name\>).PATH PATH.GET(n)</p>	<p>Returns the value of Path field of the cookie as a slash- ("/") separated list. Multiple instances of a slash are treated as single slash. If multiple Path fields are present, the value of the first instance is returned. For example, the</p>

	<p>following is a cookie with two path fields: Set-Cookie : Customer = "ABC"; PATH="/a//b/c"; PATH="/x/y/z". The following expression returns /a//b/c from this cookie: <code>http.res.set_cookie.cookie("Customer").path</code>. The following expression returns <code>b</code>: <code>http.res.set_cookie.cookie("Customer").path.get(2)</code>. Quotes are stripped from the returned value. A string of zero length is returned if the Path field or its value is absent.</p>
<p><code>HTTP.RES.SET_COOKIE.COOKIE(<name>).PATH.IGNORE_EMPTY_ELEMENTS</code> or <code>HTTP.RES.SET_COOKIE2.COOKIE(<name>).PATH.IGNORE_EMPTY_ELEMENTS</code></p>	<p>Ignores the empty elements in the list. For example, in the list <code>a=10,b=11, ,c=89</code>, the element delimiter in the list is <code>,</code> and the list has an empty element following <code>a=10</code>. The element following <code>b=11</code> is not considered an empty element. As another example, in the following expression, if a request contains <code>Cust_Header : 123,24, ,15</code> the following expression returns a value of <code>4</code>: <code>http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count</code>. The following expression returns a value of <code>5</code>: <code>`http.req.header("Cust_Header").typecast_list_t(',').count</code></p>
<p><code>HTTP.RES.SET_COOKIE.COOKIE(<name>).PORT</code> or <code>HTTP.RES.SET_COOKIE2.COOKIE(<name>).PORT</code></p>	<p>Returns the value of Port field of the cookie. Operate as a comma-separated list. For example, the following expression returns <code>80.2580</code> from <code>Set-Cookie : Customer = "ABC"; PATH="/a/b/c"; PORT= "80,2580"</code>: <code>http.res.set_cookie.cookie("ABC").port</code>. A string of zero length is returned if the Port field or value is absent</p>
<p><code>HTTP.RES.SET_COOKIE.COOKIE(<name>).PORT.IGNORE_EMPTY_ELEMENTS</code> or <code>HTTP.RES.SET_COOKIE2.COOKIE(<name>).PORT.IGNORE_EMPTY_ELEMENTS</code></p>	<p>Ignores the empty elements in the list. For example, in the list <code>a=10,b=11, ,c=89</code>, the element delimiter in the list is <code>,</code> and the list</p>

	<p>has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of</p> <p>4: <code>http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count</code>. The following expression returns a value of</p> <p>5: <code>http.req.header("Cust_Header").typecast_list_t(',').count</code></p>
<p><code>HTTP.RES.SET_COOKIE.COOKIE(<name>).VERSION</code> or <code>HTTP.RES.SET_COOKIE2.COOKIE(<name>).VERSION</code></p>	<p>Returns the value of the first Version field in the cookie as a decimal integer. For example, the following expression returns 1 from the cookie Set-Cookie : Customer = "ABC"; VERSION = "1"; VERSION = "0": <code>http.res.set_cookie.cookie("CUSTOMER").version</code>. A zero is returned if the Version field or its value is absent or if the value is not a decimal number</p>
<p><code>HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>)</code> or <code>HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>)</code></p>	<p>Returns the nth instance (0-based) of the cookie with the specified name. If the cookie is absent, returns a text object of length 0. Returns UNDEF if more than 15 Set-Cookie headers are present and the cookie is not found</p>
<p><code>HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).DOMAIN</code> or <code>HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).DOMAIN</code></p>	<p>Returns the value of the Domain field of the first cookie with the specified name. For example, the following expression returns a value of abc.com from the cookie Set-Cookie : Customer = "ABC"; DOMAIN=".abc.com"; DOMAIN=.xyz.com: <code>http.res.set_cookie.cookie("CUSTOMER").domain</code>. A string of zero length is returned if the Domain field or its value is absent</p>

<p>HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).EXPIRES or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).EXPIRES</p>	<p>Returns the nth instance (0-based) of the Expires field of the cookie with the specified name as a date string. The value can be operated upon as a time object that supports a number of date formats. If the Expires attribute is absent a string of length zero is returned.</p>
<p>HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).PATH PATH.GET(i) Or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).PATH PATH.GET(i)</p>	<p>Returns the value of the Path field of the nth cookie, as a '/' separated list. Multiple /s are treated as a single /. For example, the following expression returns /a//b/c from the cookie Set-Cookie : Customer = "ABC"; PATH="/a//b/c"; PATH= "/x/y/z": http.res.set_cookie.cookie("CUSTOMER").path. The following returns b: http.res.set_cookie.cookie("CUSTOMER").path.get(2). A string of zero length is returned if the Path field or its value is absent.</p>
<p>HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).PATH.IGNORE_EMPTY_ELEMENTS`or`HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).PATH.IGNORE_EMPTY_ELEMENTS</p>	<p>Ignore the empty elements in the list.</p>
<p>HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).PORT or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).POR</p>	<p>Returns the value or values of the Port field of the named cookie as a ',' separated list. For example, the following expression returns 80, 2580 from the cookie Set-Cookie : Customer = "ABC"; PATH="/a/b/c"; PORT= "80, 2580": http.res.set_cookie.cookie("ABC").port. A string of zero length is returned if the Port field or its value is absent.</p>
<p>HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).PORT.IGNORE_EMPTY_ELEMENTS or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).PORT.IGNORE_EMPTY_ELEMENTS</p>	<p>Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the</p>

	<p>following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of</p> <p>4: http.req.header("Cust_Header").typecast_list_t(',').ignore_empty_elements.count. The following expression returns a value of</p> <p>5: http.req.header("Cust_Header").typecast_list_t(',').count</p>
<p>HTTP.RES.SET_COOKIE.COOKIE(\<name\>, \<integer\>).VERSION or HTTP.RES.SET_COOKIE2.COOKIE(\<name\>, \<integer\>).VERSION</p>	<p>Returns the value of Version field of the nth cookie as a decimal integer. A string of zero length is returned if the Port field or its value is absent.</p>
<p>HTTP.RES.TXID</p>	<p>Returns the HTTP transaction ID. The value is a function of an internal transaction number, system boot time and system MAC address.</p>