# HTTP Service Reference

## Tagged Values

### <<E2EHTTPService>>

| Name Dialog / Tag | Description | Allowed Values / Examples | |
|---|---|---|---|
| **Port** (port) | Specifies the machine port number the service is binding to. This port number can be given at the service level only. | any number Using ports below 1024 may require additional privileges. | |
| **Trace Port** (tracePort) | In order to test the HTTP service using the Trace Analyzer, the model compiler generates a SOAP interface for each HTTP service interface. The SOAP service will be listening on this trace port. | any positive integer | |
| | | service port +40 000 | default |
| **Max Request Body Size** (maxRequestBodySize) | Runtime 2021.2 Specifies the maximum size of the request in KB (1 KB = 1024 Bytes). This can be used to prevent DoS or similar attacks. When the payload of the service exceeds the given maximum, incoming request are rejected. If you have multiple HTTP services in one composite, you can only apply the same value to all. If you specify different values on each of them, the Runtime will pick one and apply them to all. | any positive integer | |
| | | 0 | Accept unlimited requests (default of services compiled with Builder versions < 7.12.0). |
| | | **2048** | Builder 7.12.0 2MB, default if not specified |
| **Max Request Header Size** (maxRequestHeaderSize) | Runtime 2022.6 Specifies the maximum size of the request header in KB (1 KB = 1024 Bytes). This can be used to prevent DoS or similar attacks. When the header payload of the service exceeds the given maximum, incoming request are rejected. ⓘ **Compatibility Hint** For older Runtimes, a limit of 8 KB applies. If you have multiple HTTP services in one composite, you can only apply the same value to all. If you specify different values on each of them, the Runtime will pick one and apply them to all. | any positive integer | |
| | | **8** | 8 KB (default if not specified). |
| **Authentication Mode** (authenticationMode) | Defines the authentication mode to be used with this SOAP service. <br> • If HTTPBasicAuth is used, you define that the service composite authenticates the user. User name and password must then be given in the tagged value user. For more details see Basic Authentication with the E2E Bridge. <br> • If HTTPRemoteAuth is used, a revers proxy needs to authenticate the user and pass it to the service. The service then expects a set REMOTE-USER HTTP header. Typically the Apache reverse proxy is used for that. <br> Independent of the authentication mode, the user is found in the principal object if the authentication succeeded. | none | no authentication |
| | | HTTP BasicAuth | HTTP basic authentication |
| | | HTTP RemoteAuth | HTTP remote authentication |
| **User** (user) | Specifies user and password for the use of HTTP Basic authentication. | user/password | |

## HTTP Port Type Operation Parameters

| Name | Type | Direction | Description |
|---|---|---|---|
| **method** | String | in | Specifies the HTTP method. |
| **uri** | String | in | Specifies the complete URI. |

| | | | | |
|---|---|---|---|---|
| **inputContent** | Blob | in | | Specifies the HTTP content send by POST or PUT. |
| **getParameters** | Array of Parameter | in | | Specifies all GET parameters encoded in the URL as an array of Parameter classes containing name/value pairs. |
| **postParameters** | Array of Parameter | in | | Specifies all form parameters posted by the client. |
| **inputHeaderParameters** | Array of HeaderField | in | | DeprecatedThis attribute is deprecated as of Runtime 2020.11. Please use **inputHttpHeaderMap** (see below) for new implementations as its implementation complies to the HTTP specification.<br><br>Specifies all header fields as an array of HeaderField classes containing name/value pairs. |
| **pathParameters** | Map | in | | For RESTful service calls: Specifies all parameters derived from the path as a Map of name/value pairs. |
| **inputHttpHeaderMap** | Map of Entry | in | | Runtime 2020.11 Header information as a map. The map contains arrays of header value strings whereas the header name is the key of the map.<br><br>• Header names are lowercase and treated case insensitive.<br>• Multiple headers with the same name are treated as arrays.<br><br>Refer to HTTP Header Support for more information on the standard xUML HTTP headers. |
| **outputContent** | Blob | out | | Contains the content sent back. |
| **outputHeaderParameters** | Array of HeaderField | out | | Use this parameter to set the HTTP output header if necessary. |
| **httpStatusDescription** | String | out | | Contains the resulting HTTP status description. If not set explicitly using this object, the service returns OK if no exception occurred or Internal Error otherwise. |
| **httpStatusCode** | Integer | out | | Contains the resulting HTTP status code. If not set explicitly using this object, the service returns 200 if no exception occurred or 500 otherwise. |

## Parameter Types

| REST Type | Attribute | Type | Description | Values/Example |
|---|---|---|---|---|
| **Entry** | key | String | Key of the map entry. | |
| | value | Array of Any | List of values of the map entry. | |
| **HeaderField** | name | String | Name of the header field. | |
| | value | String | Value of the header field. | |
| **Parameter** | name | String | Name of the parameter. | |
| | value | String | Value of the Parameter. | |