REST Service Connection

This sets the connection, request, and response parameters for REST services. The output data of the connection is used by the REST Request node.

Connection parameters

- **Service URL**: The address to which REST service requests will be sent.
- **Method**: Select an HTTP method to use when sending requests: `GET`, `DELETE`, `POST`, `PUT`, or `PATCH`.
  - Selecting `GET` or `DELETE` methods requires the following additional settings:
    - **Allowable response content-type**: The REST Request node using this connection checks if the response type matches the one specified in this parameter. If the types do not match, the output dataset of this node will contain an error message. Select `*/*` to disable this check.
  - Selecting `POST`, `PUT`, or `PATCH` methods requires the following settings:
    - **Request content-type**: Sets the value of the `Content-type` header for the HTTP request to the REST service. You can choose from a list of predefined values or enter a custom value.
    - **Allowable response content-type**: Same as for `GET` and `DELETE` methods (see above).
- **Request parameters**: Defines a list of request parameters, with their names and data types. Subsequently, these values are inserted into the request URL within the node calling the REST service.
- **Authentication to execute request**
  - **Authentication type**: A dropdown list with authentication options:
    - **Not required**: When this option is selected, Megaldata does not perform authentication when connecting to the service.
    - **Username and password**: Select this if authentication by username/password is required. The following username and password authentication types are supported:
    - Basic
    - Digest
    - NTLM
    - Passport
    - Negotiate
    - **OAuth token**: Select this to connect to the web service using the OAuth 2.0 protocol.
    - **Client certificate from storage**: Select this for SSL authentication using a certificate. This authentication type is used for the Windows operating system.

        - **Client certificate files**: Select this if client certificate authentication is required. With this authentication type, you must select the required certificate and private key from the file store, and also enter the private key password.
    - **Username** and **Password**: These fields become available if the **Username and password** type is selected. They define the user authentication parameters for registration in the web service system.
    - **OAuth token**: This field becomes active if the **OAuth Token** type is selected. Enter the token string here.
    - **Certificate store**: A dropdown list that becomes available when **Client certificate from storage** is selected. Select one of the certificate stores available in the system. Then, select the certificate to be used for authentication in the table below.
    - **User certificate**: This field becomes available when **Client certificate files** is selected. It specifies the location (in the file store) of the certificate used to connect to the REST service.
    - **Private key**: This field becomes available when the **Client certificate files** option is selected. It specifies the location (in the file store) of the private key used to connect to the REST service.
    - **Private key password**: This field becomes active if the **Client certificate files** type is selected. This is the password to decrypt the key.
- **Connection timeout (ms)**: The maximum time in milliseconds allocated for establishing a TCP connection with the server. If exceeded, the connection will be terminated with an error status. Set this parameter to zero for unlimited timeout.
- **Data exchange timeout (ms)**: The maximum time allocated for sending an HTTP request and receiving a response. When exceeded, the exchange will be terminated with an error status. If the parameter is set to zero, the timeout is unlimited.
- **Ignore SSL certificate errors**: Mark this checkbox to ignore errors during the validation of the server certificate when connecting to the server.

Read on: SOAP Service Connection