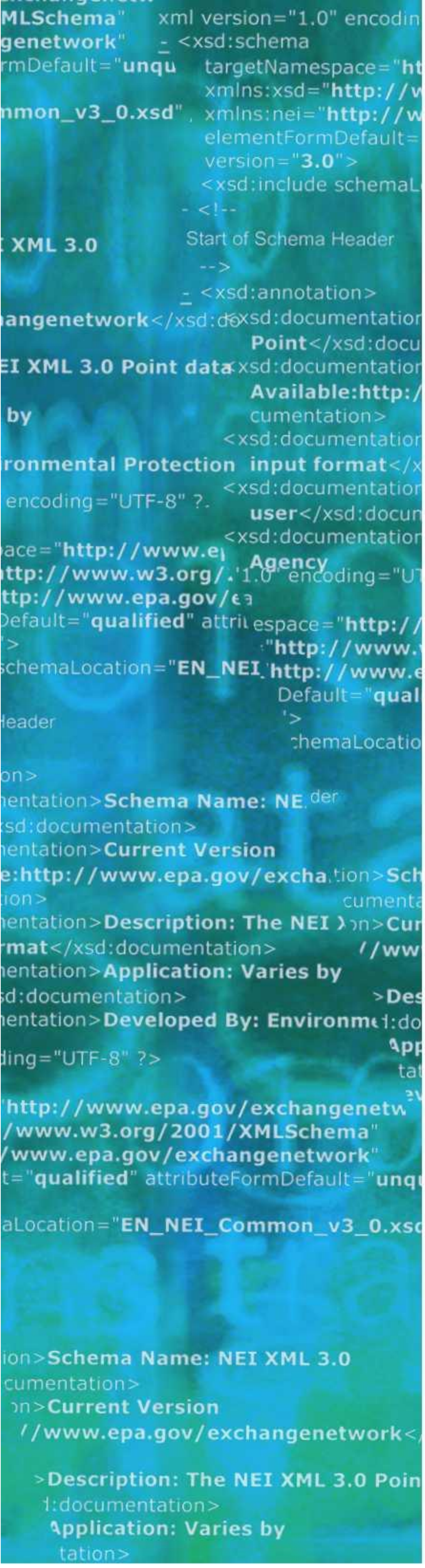


Exchange Network REST Guidance:

Optimize Your REST Services and Simplify Their Usability

Version: 1.1

May 28, 2013

Environmental Information exchange Network

Revision History

Version Number	Description of Change	Change Effective Date	Change Entered By
0.9	Initial draft	November 10, 2012	Chris Clark, Yunhao Zhang, Rob Willis, Megan Parker, Doug Timms, Kurt Rakouskas
1.0	Final approved	January 30, 2013	
1.1	Minor edits to section 6.1 to correct a typo and clarify expression of multiple values for the same parameter.	May 28, 2013	Kurt Rakouskas

Table of Contents

1 Introduction

This document provides guidance for implementing RESTful web services for environmental information. The Exchange Network strongly recommends following this guidance to simplify the discovery and consumption of REST services through standardized design, documentation and registration. These recommendations are relevant for data providers implementing REST services either as an extension of their Exchange Network Node (i.e. providing a REST interface for an existing Exchange Network Query service) or through a separate application.

This document describes how a data provider can maximize the usability of REST services by adhering to URL and documentation standards. For maximum benefit to consumers, data providers should adhere to the URL and documentation standards whenever possible and also register the services with the Exchange Network Discovery Service (ENDS). If a REST service provider is unable or chooses not to use the URL standard, it is still beneficial to implement the standardized documentation and service registration.

This document provides implementation guidance for four distinct scenarios:

Scenario	How this Document Helps	See Section(s)
1. **You want to *create* a new REST service that is standardized, well documented, discoverable, and easy to use.**	The Network's standard REST service URL format, documentation standards, and service registration can help your users more easily discover your services and generate requests for data.	**3, 4, and 5**
2. **You have created or are about to create a REST service and would like to *document and register it* so people can discover and use it more easily**.	Following the Network's documentation standards along with the instructions for registering your service in the Exchange Network Discovery Service (ENDS) can help your users more easily find and use your data. ENDS registration also makes your service automatically discoverable by Exchange Network data access tools.	**4 and 5**
3. **You have created or are about to create a REST service and would like to *document* it so people can use it.**	If the Network's recommended URL format does not meet your business needs, you can follow our documentation standards to give your users a consistent and complete set of information that will help them understand how to use your service.	**4**
4. **You have an existing Exchange Network Query service and you want to *make your Query service available via REST* without any coding**.	The Exchange Network offers a free tool that can expose your existing SOAP-based Query services as a REST endpoint. This document provides background on this REST-Proxy tool and instructions on how to use it.	**6**

Table 1: Scenarios Addressed in this Document

2 Background

2.1 What is REST?

Representational State Transfer (REST) is a software architectural style commonly used in the construction of Web-based applications and has emerged as a predominant Web service design model. REST is neither a specification nor a technology. It is a design approach whereby any resource can be represented by a URL (Uniform Resource Locator). Accessing a URL will result in a resource being transferred to the requester.

2.2 REST and the Exchange Network

The Exchange Network (EN) has created a powerful framework for exchanging environmental information between partners. To date, many of the Network's data exchanges were built to enable automated reporting and submission of regulatory information. These data exchanges were built using web services based on a technology called Simple Object Access Protocol (SOAP) that is well suited for securely exchanging large payloads of structured information or for synchronizing databases.

The Exchange Network governance is actively seeking to promote another exchange pattern based on data publishing. When partners publish data, they make it available through a web service so that other partners or applications can easily discover, request, and access the data on demand. Unlike data submissions, which are initiated by data providers, published data is requested and used by a data consumer on demand. The Exchange Network's current model asks that EN partners implement a SOAP-based Query service. In many scenarios, however, a RESTful web service interface may offer a more appropriate solution. REST offers a clean, simple and easy-to-understand method of retrieving information. There is no special software to install. A web browser, a web application or a mobile app can be used to access the service directly. In addition, users can take advantage of Exchange Network tools that simplify the generation of complex URLs and service requests.

REST is immediately accessible to internet users and offers some significant advantages in delivering data to applications that run on devices such as smart phones and tablets using formats like XML or JavaScript Object Notation (JSON).

In summary, adding REST interfaces to Exchange Network services will help broaden their reach to a much wider audience while simplifying the discovery and consumption of environmental data. Users will be able to discover and access services seamlessly without regard to the underlying technology (SOAP or REST). In situations where they are appropriate, REST services can serve as a useful compliment to the Exchange Network's existing set of SOAP-based services and help to more rapidly expand data sharing among Network partners.

2.3 Recommended Constraints on the Use of REST Services in the Exchange Network

The simplicity of RESTful services as compared to SOAP-based services can make them an attractive choice for providing access to information. However, data providers must weigh the relative strengths and weaknesses of each approach and pick the one that is best matched to the business needs.

At this time, the Exchange Network recommends that developers only consider using REST services under the following conditions:

- **Business needs can be met using synchronous queries**
  The HTTP GET method is well suited for supporting simple queries that do not require multi-step processing. Business processes that require behind the scenes status checks or complex data extractions before making results available are typically better served by the asynchronous behavior of EN Nodes using the SOAP-based Solicit method. Similarly, data submission services that include status checking and downloading of results may also be poor candidates for the REST approach.

- **Data are considered completely public and no security is required**
  There are fewer standard security specifications with REST services than with SOAP-based web services. The Exchange Network recommends using REST services only for data that are considered public. Exchange Network partners should use the EN's SOAP-based web services in conjunction with the Network Authentication and Authorization Service (NAAS) for any data that require restricted access.

In the future, the Exchange Network may produce guidance on constructing RESTful web services for business processes that use other HTTP methods including POST, PUT, DELETE, etc.

3 Exchange Network REST URL Standard

REST web services offer a very useful approach for sharing data, but they are not always simple and straightforward to use. Because REST is a style and not a technology specification, there is a great deal of variation in how developers implement their services. There is no standard approach for creating a REST service, so developers are left to choose their own URL structures and their own means of naming and delimiting query parameters. This flexibility is one of REST's strengths, but it can also make life challenging for other developers or consumers that want to use those services to access data. URLs can quickly become complex and users may struggle to understand exactly how to structure their requests and retrieve the correct information.

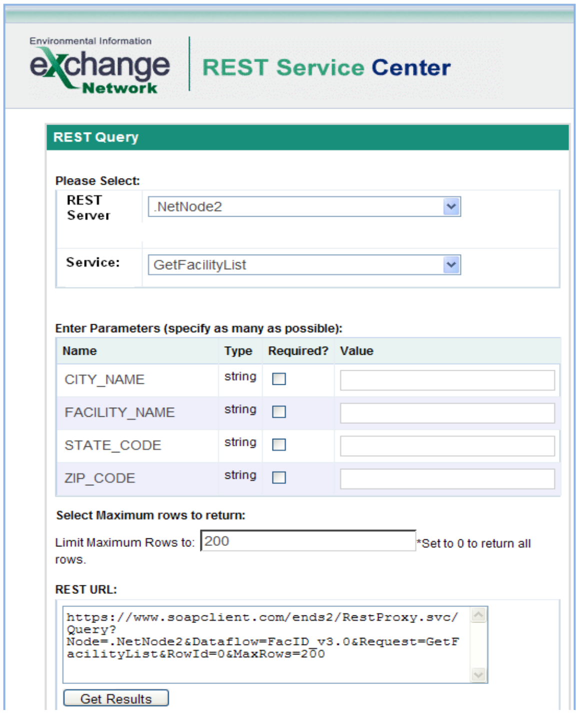
The Exchange Network believes that there is value in encouraging standardization around certain elements of the URL structure. By following a standard, REST service developers can help users more easily understand and use their services. Standardization can be particularly helpful when users are aggregating or mashing-up data of different types or from different sources. Standardizing can also open up possibilities to use shared tools that can help automate the process of building service requests (see example to the right).

Maximizing Benefit: Exchange Network REST Service Center

The Exchange Network REST Service Center is a website that helps consumers more easily make requests against REST services. It works for any REST services that are:

- Consistent with the EN REST URL and documentation standards, and
- Registered in ENDS

The website uses the information documented in ENDS to guide users through the creation of a customized service request. Users can request data using the site, or copy and paste the custom URL for use in another application.



Please Select:

REST Server: .NetNode2

Service: GetFacilityList

Enter Parameters (specify as many as possible):

Name	Type	Required?	Value
CITY_NAME	string	□	
FACILITY_NAME	string	□	
STATE_CODE	string	□	
ZIP_CODE	string	□	

Select Maximum rows to return:

Limit Maximum Rows to: 200 *Set to 0 to return all rows.

REST URL:

```
https://www.soapclient.com/ends2/RestProxy.svc/
Query?
Node=.NetNode2&Dataflow=FacID_v3.0&Request=GetF
acilityList&RowId=0&MaxRows=200
```

3.1 Standard REST URL Structure

The basic structure of the standardized EN REST URL is:

Server/Service/ServiceParameters

These three URL components are further described here:

URL Component	Description
1. Server	A URL endpoint where one or more REST services exist. Typically, the endpoint URL contains a protocol identifier (e.g. http://), a domain name (e.g. www.example.gov), and may optionally include a portion of

	the URL that is not service specific.
2. Service	Identification of a particular service on a server. **One or more Services may exist per Server.**
3. Service Parameters	Parameters that can be included in the service request, to convey operating instructions for the service to execute, such as filtering data to return. **Zero or more Service Parameters may exist per Service.**

3.2 Standard REST URL Formatting Rules

In addition to adhering to the standard URL components, the following formatting rules pertain to the structure for standardized EN REST URL Endpoints:

- **Rule 1: Service Parameter Delimiting:** Service parameters are to be delimited using the ampersand (&) as shown in the example given in section 3.3. Any other parameter delimiters will create difficulties for any service automation tools.

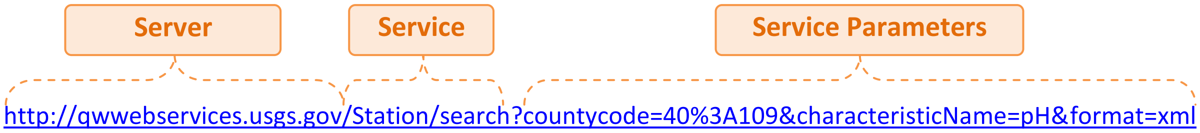
- **Rule 2: Explicit Parameter Naming:** Service parameters are to be formatted as name/value pairs, as shown in the example in Section 3.3. Any URLs that omit the parameter name (and therefore only include parameter values in a specific sequence) will create difficulties for any service automation tools. This approach has the added benefit of allowing requesters to provide parameters in any order, and not be bound to unintuitive parameter ordering requirements.

- **Rule 3: Predefined Parameter Names:** The Exchange Network has created several predefined parameter names for commonly used features. These predefined parameter names are:

Parameter Name	Description
format	Identifies the format in which data will be returned. Commonly used formats should use the following naming convention (xml, json, html, csv, xls). The returned format is xml if the parameter is omitted.
rowId	When returning paged results, identifies the starting record number to be retrieved. The default value of rowId is 0.
maxRows	Identifies the maximum number of records to return. Service developer should consider supporting the maxRows parameter if the anticipated size of the response is large. All records are to be returned if maxRows parameter is not supplied.

3.3 Example REST Endpoint

This example REST service endpoint illustrates use of the EN REST URL formatting rules.



http://qwwebservices.usgs.gov/Station/search?countycode=40%3A109&characteristicName=pH&format=xml

Notes:

- http://qwwebservices.usgs.gov is part of the "Server" hierarchy because it will be the same for all different REST services deployed at that endpoint.
- Station/search? Is part of the "Service" hierarchy because it will always be the same anytime someone calls the Station search service, *but* will have a different value if another REST service was available at the same server (e.g. http://qwwebservices.usgs.gov/Projects/search?)

4 Exchange Network REST Service Documentation Standard

REST web services must be well documented so that users can easily understand how to construct requests and retrieve the data that they are seeking. The Exchange Network recommends that data providers document certain standard information so that services are as usable as possible. While the Exchange Network encourages data providers to use the URL format in Section 3 of this document, there are many different ways of formatting a URL. These documentation standards should be used whether or not a provider chooses to use the standard URL format.

Information Availability: Information should be published in a location that is accessible by potential service users. This could be a public web page (e.g. http://qwwebservices.usgs.gov/technical-documentation.html) or the Exchange Network Discovery Service (See Section 5).

This section will use an example REST endpoint for illustrative purposes:
http://qwwebservices.usgs.gov/Station/search?countycode=40%3A109&characteristicName=pH&format=xml

4.1 Information to Document

This section outlines the information that should be documented, categorized by hierarchy level:

Hierarchy	Name	Description	Example[1]
Server	Address	The HTTP root URL where the service will be provided. In most situations, it should contain only the protocol and domain name without virtual directories. It will be used to construct absolute URL string for REST services.	http://qwwebservices.usgs.gov
Server	Server Name	The name of the REST server. It should be brief and meaningful.	NWIS-WQ
Server	Server Description	A description of the REST server.	National Water Information System (NWIS) Water-Quality Web Services
Server	Status	The status of the service. Select "Operational" if the server is available, otherwise choose "Offline". Assumed Operational if not supplied.	Operational
Server	Provider	The name of the organization that provides the services.	USGS
Server	Contact	Email addresses for technical contacts.	johndoe@usgs.gov
Service	Service URL Template	This is a relative URL without the Server Address part. It should contain everything needed by the server except the base server address and	/Station/search?

[1] Based on example service:
http://qwwebservices.usgs.gov/Station/search?countycode=40%3A109&characteristicName=pH&mimeType=xml

Hierarchy	Name	Description	Example[1]
		parameters.	
Service	Service Name	A short name of the service. It should be unique within the server and the name should only contain alphanumeric characters without white spaces. This will be used as the unique identifier of the service.	Station
Service	Description	A more detailed description of the service.	Returns listing of stations in WQX XML format that match given data filters.
Service	Shared Option	This is a flag that determines how the service will be published. A private service will not be published, a shared service will be published to only authenticated users, and a public service will be viewable by any users.	Public
Service	HTTP Method	Identifies which http method is expected for the request. "GET" will be assumed by default if this is not specified.	GET
Service Parameter	Name	Name of parameter as it would appear in query string	CharacteristicName
Service Parameter	Description	A human readable description of the parameter.	One or more case-sensitive characteristic types (groupings)
Service Parameter	Type	A classification of the datatype of the parameter.	String
Service Parameter	Required Indicator	Indicates whether the parameter must be supplied.	False
Service Parameter	Allow Multiple Indicator	For query parameters, indicates whether multiple values are allowed	True
Service Parameter	Multiple Separator	Indicates which character is used to separate multiple values.	;
Service Parameter	Allowable Values	Indicates supported, allowable values for the parameter.	pH

Table 2: Information to Document When Describing REST services

4.2 Additional Documentation Expectations

1. **Example Requests:** Provide one or more example requests to illustrate for users how services must be called.
2. **Expected Response:** A description of the response users should expect to receive (e.g. "Response will be in WQX v2.1 XML format")
3. **Expected Error Response:** A description of the response users should expect when error conditions are encountered.

5 Registering Your REST Service in ENDS

Section 4 describes the information that should be documented when fully describing REST services. However, describing the services does not guarantee maximum exposure / communication of your service to potential users. The Exchange Network recommends that providers register REST services in the Exchange Network Discovery Service (ENDS) to ensure your service reaches the widest audience.

ENDS is a central repository of environmental data publishing web services, and was created to allow for the discovery of a wide variety of environmental data in a central location. For more information, refer to http://www.exchangenetwork.net/exchange-network-discovery-service-ends/.

ENDS supports online registration of both Node services (SOAP) and REST services. This section describes how to register REST services in ENDS, using the standard documentation described in the previous section.

5.1 Accessing ENDS

ENDS offers two websites for administrators to register and manage SOAP and REST services. They are located at:

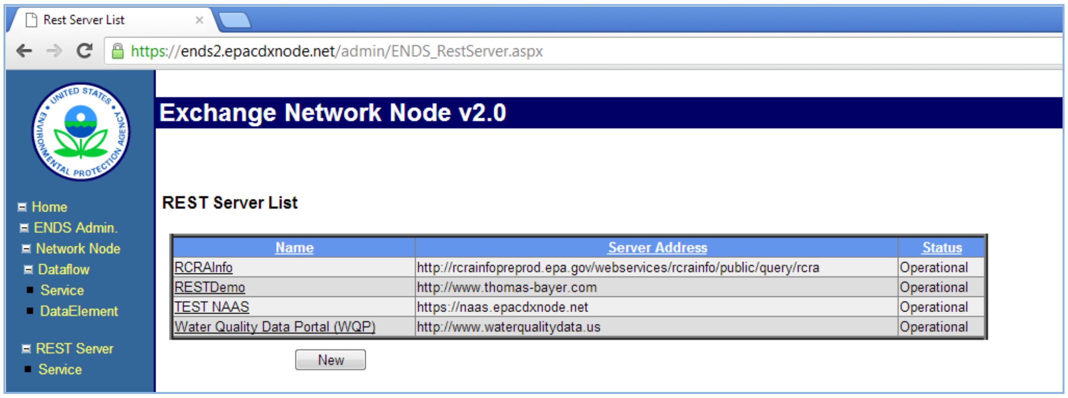
- **TEST:** https://ends2.epacdxnode.net/admin/default.aspx
- **PRODUCTION:** https://ends2.epa.gov/admin/default.aspx

These websites are for node administrators only; a user may need to be authorized in order to access the web pages. Please contact the node helpdesk at nodehelpdesk@epacdx.net if you need assistance.

5.2 Registering a REST Server

A service could not exist without a server. So, a REST server must be defined before a description of services can be provided. A server is an endpoint where various services exist. The information to be registered here corresponds with the "Server" documentation standards identified in Section 4.

You will be prompted to login using your NAAS User ID and password. After logging in, you should see a link for 'REST Server' on the left panel. When clicked, a list of existing REST servers will be displayed as shown here:

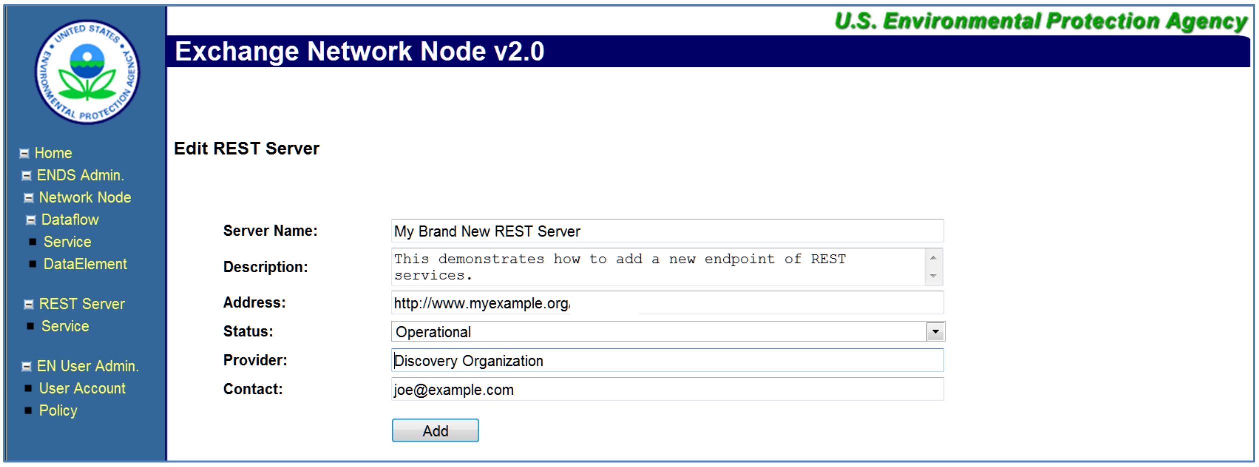


REST Server List

Name	Server Address	Status
RCRAInfo	http://rcrainfopreprod.epa.gov/webservices/rcrainfo/public/query/rcra	Operational
RESTDemo	http://www.thomas-bayer.com	Operational
TEST NAAS	https://naas.epacdxnode.net	Operational
Water Quality Data Portal (WQP)	http://www.waterqualitydata.us	Operational

REST Server Listing in ENDS

Click on the 'New' button to add a new REST server to the list. A web form as shown below will be displayed:



Edit REST Server

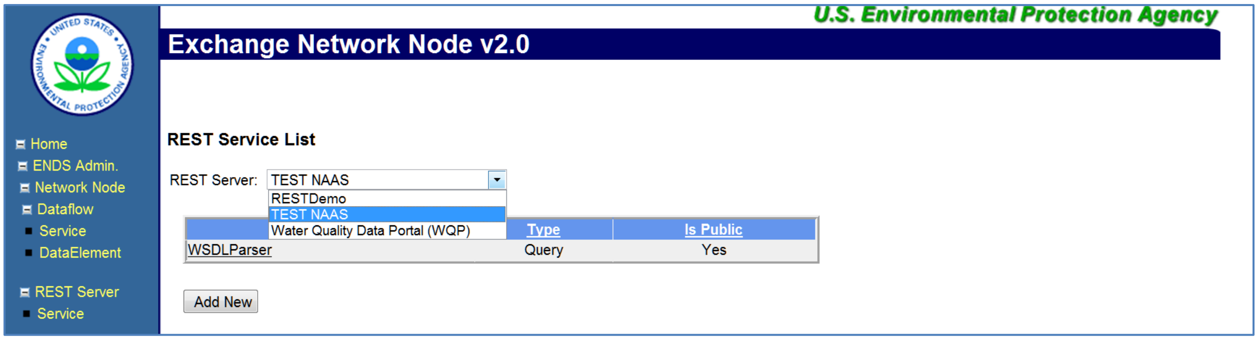
Server Name:	My Brand New REST Server
Description:	This demonstrates how to add a new endpoint of REST services.
Address:	http://www.myexample.org
Status:	Operational
Provider:	Discovery Organization
Contact:	joe@example.com

Edit REST Server Information

5.3 Registering REST Services

After creating a REST server, you must associate new services with the server. The information to be added here corresponds with the "Service" documentation standards identified in Section 4.

Click on the Service link on the left panel, which will display a list of REST servers and services as shown here:



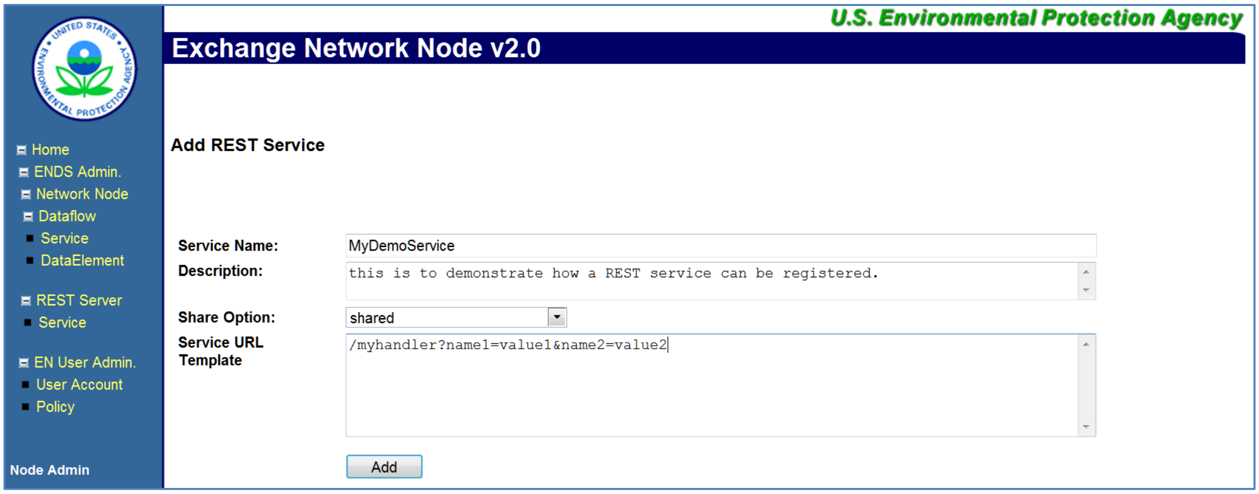
REST Service List

REST Server: TEST NAAS

	Type	Is Public
WSDLParser	Query	Yes

REST Service Listing

First select the REST server, and then click on the Add New button to add a new REST service.



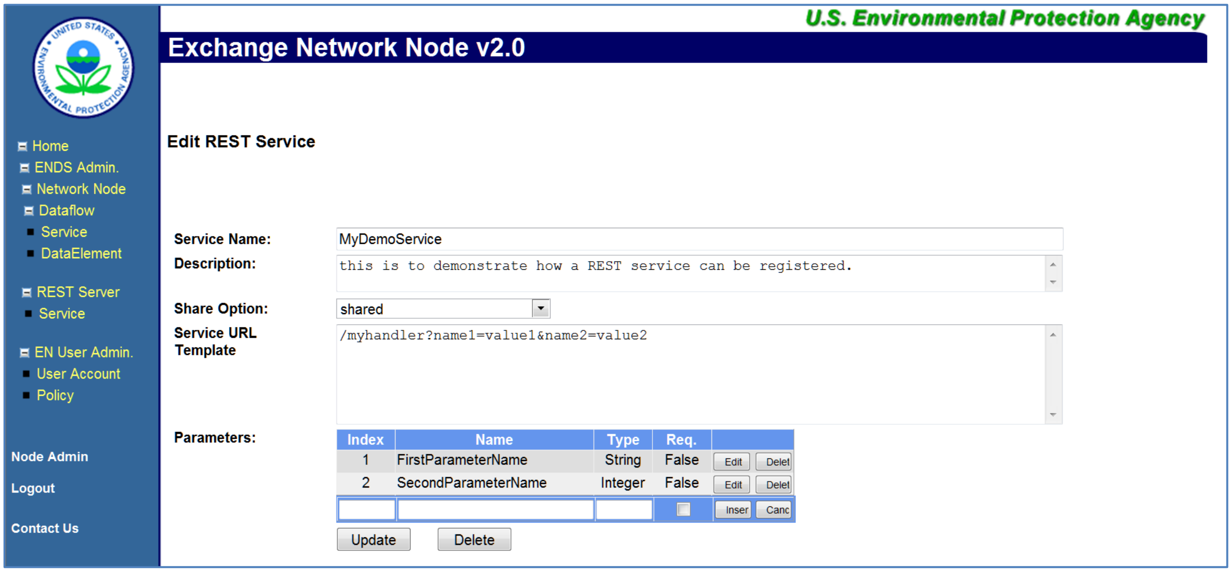
Add REST Service

Service Name:	MyDemoService
Description:	this is to demonstrate how a REST service can be registered.
Share Option:	shared
Service URL Template	/myhandler?name1=value1&name2=value2

Edit REST Service Details

5.4 Registering REST Service Parameters

The service registration supports only REST services using name-value pairs as the parameter format. Service parameters can be added after a service is created as shown in the following screenshot:



Edit REST Service

Service Name:	MyDemoService
Description:	this is to demonstrate how a REST service can be registered.
Share Option:	shared
Service URL Template	/myhandler?name1=value1&name2=value2

Parameters:

Index	Name	Type	Req.	
1	FirstParameterName	String	False	Edit Delet
2	SecondParameterName	Integer	False	Edit Delet
			□	Inser Canc

Edit REST Service Details

Given the service definition, the URL for accessing the service will have the following form:

http://myexample.org/myhandler?name1=value1&name2=value2&FirstParameterName=firstParamValue&SecondParameterName=secondParamValue

After registering your service in ENDS, you may test it using the Exchange Network REST Service Center described in section 3. The REST Services Center is located at: https://nodewebrss.epa.gov/RestBuilder

5.5 Mapping of REST Documentation Items to ENDS XML

The following table identifies the XPath in the ENDS XML file for each REST documentation item:

Hierarchy	Name	ENDS XPath
Server	Address	NetworkNodeDetails/NodeAddress
Server	Server Name	NetworkNodeDetails/NodeIdentifier
Server	Server Description	NetworkNodeDetails/NodeName
Server	Status	NetworkNodeDetails/NodeStatus
Server	Provider	NetworkNodeDetails/OrganizationIdentifier
Server	Contact	NetworkNodeDetails/NodeContact
Service	Service URL Template	NetworkNodeList/Service/ServiceDocumentURL
Service	Service Name	NodeServiceList/Service/ServiceIdentifier
Service	Description	NodeServiceList/Service/ServiceDescription
Service	Shared Option	NodeServiceList/Service/Ownership
Service	HTTP Method	NodeServiceList/Service/ServiceProperty/ProperyName
Service Parameter	Name	NodeServiceList/Service /Parameter/@ParameterName
Service Parameter	Description	Described in DEDL: DataElementList/DataElement/Description
Service Parameter	Type	NodeServiceList/Service /Parameter/@ParameterType
Service Parameter	Required Indicator	NodeServiceList/Service /Parameter/@ParameterRequiredIndicator
Service Parameter	Allow Multiple Indicator	Described in DEDL: DataElementList/DataElement/DataConstrains/AllowMultiSelect
Service Parameter	Multiple Separator	Described in DEDL: DataElementList/DataElement/Properties/
Service Parameter	Allowable Values	Described in DEDL: DataElementList/DataElement/ElementValue

Table 3: Mapping of REST Documentation Items to ENDS XML Structure

6 Using the Exchange Network REST Proxy to Expose Existing Exchange Network Query Services as REST services

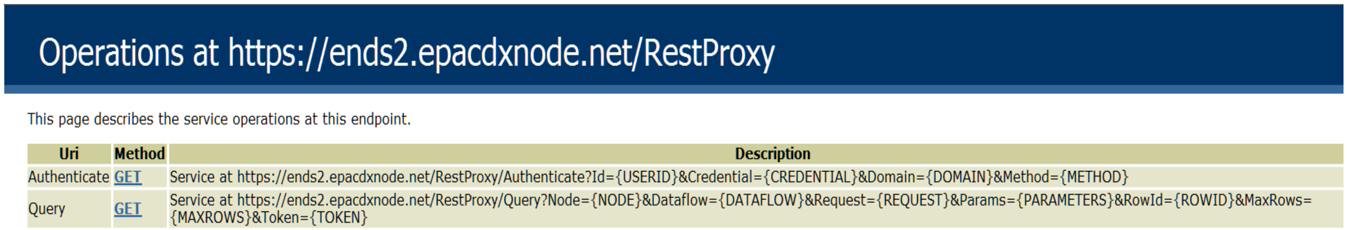
EPA's CDX offers a service called the Exchange Network REST Proxy (ENRP), which can transform existing Node services into REST services without any additional development effort from Exchange Network partners. If you already publicly publish data on your Exchange Network Node using the Query web service and the Query service is registered in ENDS, you can make those services available through a REST URL by using the REST Proxy. The REST Proxy takes the basic information used to document a Query service in ENDS and auto-generates a REST URL. When a user calls the REST service, the ENRP converts the incoming request into a standard Exchange Network web service call and then provides the data back to the requester. The URL template of the auto-generated REST endpoints is shown below:

https://{nodedomain}/{RestProxy}/{method}?Node={node}&Dataflow={dataflow}&Request={request}&Params={parameters}&RowId={rowId}&MaxRows={maxRows}&Token={token}

The root URL of the REST Proxy is:

- **TEST:** https://ends2.epacdxnode.net/RestProxy
- **PRODUCTION:** https://ends2.epa.gov/RestProxy

The format for accessing URL is shown below:



Operations at https://ends2.epacdxnode.net/RestProxy

This page describes the service operations at this endpoint.

Uri	Method	Description
Authenticate	GET	Service at https://ends2.epacdxnode.net/RestProxy/Authenticate?Id={USERID}&Credential={CREDENTIAL}&Domain={DOMAIN}&Method={METHOD}
Query	GET	Service at https://ends2.epacdxnode.net/RestProxy/Query?Node={NODE}&Dataflow={DATAFLOW}&Request={REQUEST}&Params={PARAMETERS}&RowId={ROWID}&MaxRows={MAXROWS}&Token={TOKEN}

REST Proxy

Authentication: For public services that require no authorization, the security token parameter is supplied automatically by the REST Proxy. The REST Proxy uses a public account,

public@epacdxnode.net

to obtain a security token behind the scene in this case. The account only has access to public services registered in ENDS (setting Share Option to public), and the node administrator must authorize the account for service access.

Since the Exchange Network is a secure network that requires authentication, accessing node services that require authorization is a two-step process:

Step 1: User authentication to obtain security token

Step 2: Perform an operation using the security token obtained from Step 1.

Although it is possible to perform other node operations such as Submit and Download, the main usage scenario is data publishing through the Query method at this time.

For example, to access the Query_v1.0 service of NCT dataflow in the .NetNodeTest, the REST URL would be:

https://ends2.epacdxnode.net/RestProxy/Query?Node=.NetNodeTest2.0&Dataflow=NCT&Request=Query_v1.0&Params=stringParameter|string&RowID=0&MaxRows=100

Performing the same operation against the Virtual Node should have a similar URL structure:

https://ends2.epacdxnode.net/RestProxy/Query?Node=VirtualNode&Dataflow=NCT&Request=Query_v1.0&Params=stringParameter|string&RowID=0&MaxRows=100

The REST Proxy operates based on service definitions in ENDS, which means that RESTful access through the REST Proxy works only when the service is defined in ENDS.

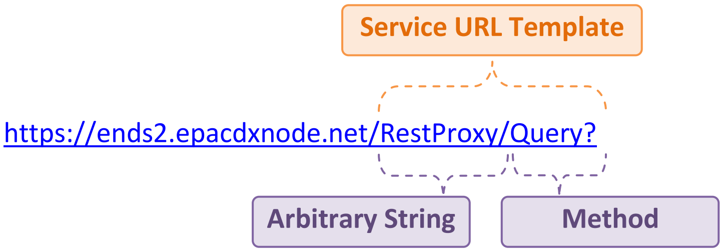
6.1 Expected Endpoint URL When Using REST Proxy

When using the REST Proxy Tool, the URL endpoint will be automatically formatted according to the convention outlined in this section.

Specific Formatting of Service URL Template: The Service URL Template will be formatted as follows:

arbitraryString/MethodName?

Where MethodName corresponds to the Exchange Network Node method (e.g. Query or Solicit), as illustrated in the following example:



Predefined Parameter Names: The following predefined parameter names may be used, which convey a specific meaning within the Exchange Network community:

Name	Description
Node	Identifies the unique Exchange Network Node ID with which this REST service is associated. If using ENDS, this should correspond with the Node ID registered in ENDS.
Dataflow	The name of the dataflow associated with the service.
Request	The name of the service request.
Params	A concatenated listing of all parameters for the Query or Solicit service in the format of: *param1|value1;param2|value2* Where parameter name/value pairs are separated with a pipe (|) delimiter and parameters are separated with a semi-colon (;) delimiter. When passing multiple values for the same parameter, the values are expressed as distinct name/value pairs. For example, ZipCode|20635;ZipCode|20637 *Note this is a condensation of all Query parameters into one REST parameter.*
RowId	An optional rowId parameter for the Query/Solicit method that identifies the starting record number to be retrieved when returning paged results.
MaxRows	An optional maxRows parameter for the Query/Solicit method that identifies the maximum number of records to return.
Token	Security token, typically referring to a NAAS security token for an authenticated NAAS user.