WRM Server REST API

Main Author

NAME	E-MAIL ADDRESS
Juho Kautiainen	juho.kautiainen@wapice.com

Table of Contents:

i List of Figures

ii List of Tables

1 INTRODUCTION

This document describes the REST API interfaces of WRM server for integration with other systems.

1.1 WRM System Overview

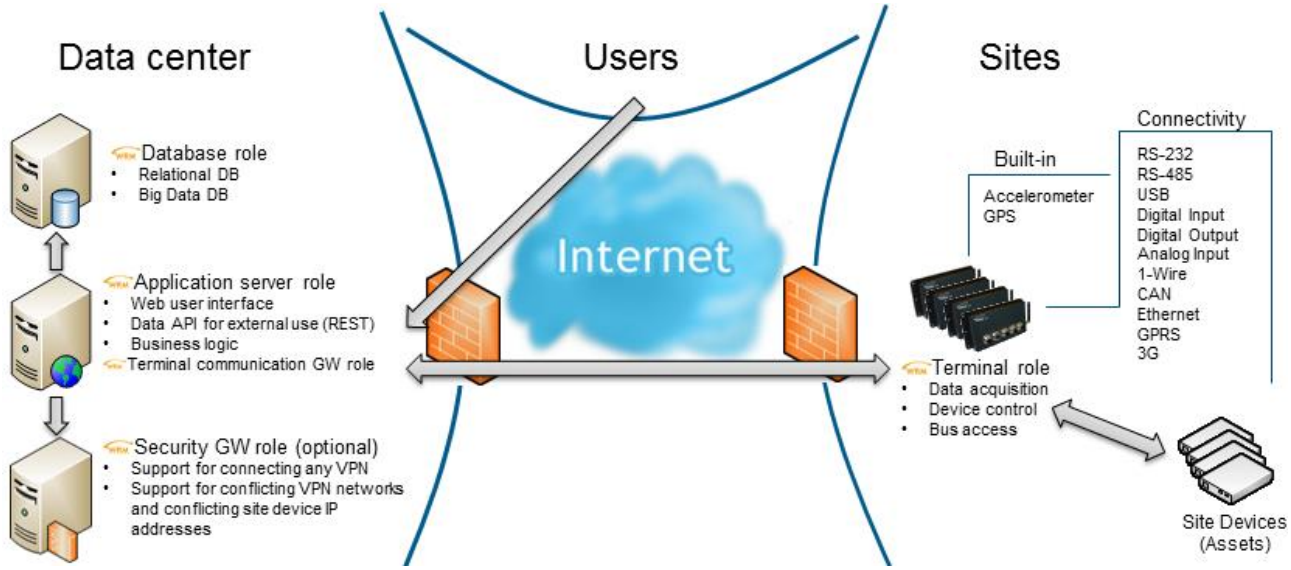


Figure 1 - WRM System Overview

1.2 REST

REpresentational State Transfer (**REST**) is a style of software architecture for distributed systems such as the World Wide Web. REST has emerged over the past few years as a predominant Web service design model.

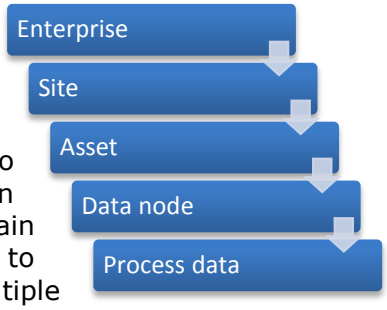
Key goals of REST include:

- Scalability of component interactions
- Generality of interfaces
- Independent deployment of components
- Intermediary components to reduce latency, enforce security and encapsulate legacy systems

An important concept in REST is the existence of resources (sources of specific information), each of which is referenced with a global identifier (e.g., a URI in HTTP). In order to manipulate these resources, *components* of the network (user agents and origin servers) communicate via a standardized interface (e.g., HTTP) and exchange *representations* of these resources (the actual documents conveying the information).

Any number of *connectors* (e.g., clients, servers, caches, tunnels, etc.) can mediate the request, but each does so without *"seeing past"* its own request (referred to as "layering," another constraint of REST and a common principle in many other parts of information and networking architecture). Thus, an application can interact with a resource by knowing two things: the identifier of the resource and the action required—it does not need to know whether there are caches, proxies, gateways, firewalls, tunnels, or anything else between it and the server actually holding the information. The application does, however, need to understand the format of the information (*representation*) returned, which is typically an HTML, XML or JSON document of some kind, although it may be an image, plain text, or any other content.

1.3 Data model



Following diagram illustrates the WRM system data model that is visible to clients through this REST API.

Root level in this model is enterprise which represents a company or some other type of customer. Enterprises can contain multiple sites. Sites represent a location of interest which users want to monitor and/or control, for example a factory or a ship. Site can also be used as a fleet to contain multiple vehicles. Sites can contain multiple assets. Assets are the resources a customer wants to monitor, for example a truck or an engine. Assets can contain multiple data nodes. Assets can also contain other assets to create a tree structure of assets to further model the site. Data nodes define what is measured from the asset it belongs to, like temperature or RPM. Data nodes then can contain multiple process data values.

Enterprise 1-N Site 1-N Asset 1-N Data Node 1-N Process data

Figure 2 - WRM data model illustration

1.4 General information

The base URL for all these interfaces is *{WRM_SERVER_ADDRESS}/rest/v{VERSION_NUMBER}* for example https://demo.wrm247.com/rest/v1/datanodes will list data nodes as explained in next chapter. Version is defined in the URL to keep compatibility with old client applications when changes to the interface are made. Each interface supports JSON or XML data. Data format can be defined by using the *format* -parameter in the request URL, e.g. https://demo.wrm247.com/rest/v1/datanodes?format=XML will return data nodes in XML format.

1.5 Authentication

This API uses HTTPS protocol to secure the communication between server and clients. HTTP Basic authentication method with username and password is used over the HTTPS protocol to authenticate users. WRM system user profiles define system level access to the data requested through this API.

2 REST INTERFACES, VERSION 1

2.1 Get Enterprises

URL: /enterprises
HTTP method: GET

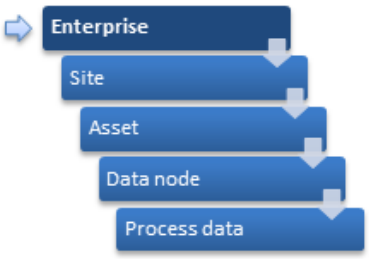


Table 1 - URL Parameters for Get Enterprises

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/enterprises?callback=foo returns: foo(response data);
format	Format (json or xml) default is json	/enterprises?format=xml
identifiers	comma separated enterprise identifiers	/enterprises?identifiers=20,30,40
name	enterprise name	/enterprises?name=wapice
expand	comma separated object field names to expand	/enterprises?expand=name

Table 2 - Response to Get Enterprises

Field	Description
href	URL to this resource
offset	How many results to skip from beginning
limit	Amount of results per page
first	URL to first page
previous	URL to previous page
next	URL to next page
last	URL to last page
items	Array of Enterprise objects with "href" element expanded by default (element expansion can be changed by using the "expand" parameter)

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/enterprises?format=json


```
{
 "href":"https://demo.wrm247.com/rest/v1/enterprises?format=json",
 "expandable":"id,name,sites",
 "offset":0,
 "limit":10,
 "next":"https://demo.wrm247.com/rest/v1/enterprises?format=json&offset=10",
 "last":"https://demo.wrm247.com/rest/v1/enterprises?format=json&offset=10",
 "fullSize":14,
 "items":[
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/601"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/600"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/300"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/550"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/250"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/602"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/500"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/650"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/552"},
           {"href":"https://demo.wrm247.com/rest/v1/enterprises/551"}
 ]
}
```

2.2 Enterprise Object

URL: /enterprises/123 *(where 123 is globally unique id for the enterprise)*
HTTP method: GET

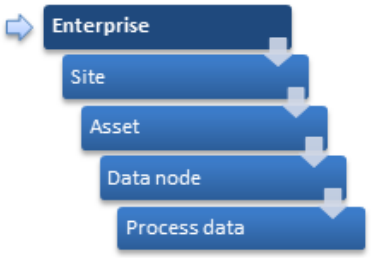


Table 3 - URL Parameters for Enterprise Object

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/enterprises/123?callback=foo returns: foo: foo(response data);
format	Format (json or xml) default is json	/enterprises/123?format=xml
expand	Comma separated field names to expand	/enterprises/123?expand=sites.name

Table 4 - Response to Enterprise Object

Field	Description
href	URL to this resource
name	User friendly name of the enterprise
identifier	Unique identifier for Enterprise. This can be used for query filtering when searching for enterprises, sites, assets or data nodes.
sites	Object containing URL to the sites of this enterprise

Table 5 - Enterprise Object Sub-URLs

URL	Description	Supported query parameters
/enterprises/{enterpriseId}/sites	list sites under the specified enterprise	name callback format expand offset limit

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/enterprises/551?format=json


```
{
"href":"https://demo.wrm247.com/rest/v1/enterprises/551?format=json",
"expandable":"sites.assets,sites.dataNodes,sites.enterprise,sites.id,sites.model
,sites.name,sites.parameters",
"id":551,
"name":"Some Enterprise",
"sites":{
  "href":"https://demo.wrm247.com/rest/v1/enterprises/551/sites"
  }
}
```

2.3 Get Sites

URL: /sites
HTTP method: GET

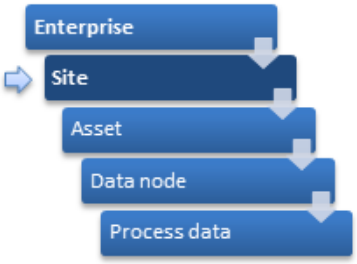


Table 6 - URL Parameters for Get Sites

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/sites?callback=foo returns: foo(response data);
format	Format (json or xml) default is json	/sites?format=xml
parameters	Comma separated array of site parameter and value pairs surrounded with square brackets.	/sites?parameters=[someParam=true, someOtherParam=124]
enterprises	comma separated enterprise identifiers	/sites?enterprises=WAPICELTD,SOMECOMPANY
identifiers	comma separated site identifiers	/sites?enterprises=WAPICELTD&identifiers=SITE1,SITE2
name	site name	/sites?enterprises=WAPICELTD&name=Site1
expand	comma separated object field names to expand	/sites?enterprises=WAPICELTD&identifiers=SITE1,SITE2&expand=items.name

Table 7 - Response to Get Sites

Field	Description
href	URL to this resource
offset	How many results to skip from beginning
limit	Amount of results per page
first	URL to first page
previous	URL to previous page
next	URL to next page
last	URL to last page
items	Array of href elements containing URLs to the returned sites

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/sites?format=json


```
{
 "href":"https://demo.wrm247.com/rest/v1/sites?format=json",
 "expandable":"assets,dataNodes,enterprise,id,model,name,parameters",
 "offset":0,
 "limit":10,
 "next":"https://demo.wrm247.com/rest/v1/sites?format=json&offset=10",
 "last":"https://demo.wrm247.com/rest/v1/sites?format=json&offset=10",
 "fullSize":18,
 "items":[
          {"href":"https://demo.wrm247.com/rest/v1/sites/64675"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/50600"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/50663"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/45"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/74457"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/223"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/50257"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/35593"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/69463"},
          {"href":"https://demo.wrm247.com/rest/v1/sites/8"}
  ]
}
```

2.4 Site Object

URL: /sites/123 *(where 123 is globally unique id for the site)*
HTTP method: GET

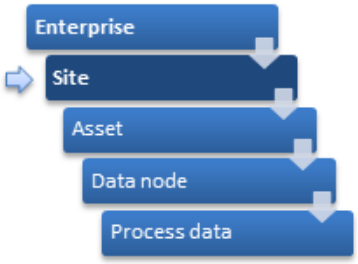


Table 8 - URL Parameters for Site Object

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/sites/123?callback=foo returns: foo: foo(response data);
format	Format (json or xml) default is json	/sites/123?format=xml
expand	Comma separated field names to expand	/sites/123?expand=enterprise

Table 9 - Response to Site Object

Field	Description
href	URL to this resource
name	User friendly name of the site
identifier	Unique identifier for Site in Enterprise context. This can be used for query filtering when searching for sites, assets or data nodes.
model	Site model name
parameters	Parameters contain meta-data of the site. Objects containing parameters and their values inside an array: *[{"parameter":"example1","value":"exampleValue1"},* *{"parameter":"example2","value":"exampleValue2"},…]*
dataNodes	Object containing URL to the data nodes that belong to this site. Note that the resource behind the URL only displays the data nodes created directly under the site and not the ones under the assets in that site.
assets	Object containing URL to the assets that are under this site in site hierarchy
enterprise	Object containing URL to the enterprise this site belongs to

Table 10 - Site Object Sub-URLs

URL	Description	Supported query parameters
/sites/{siteId}/assets	list assets under the specified site	name callback format expand offset limit

/sites/{siteId}/datanodes	lists datanodes under the specified site	name callback format expand offset limit

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/sites/64675?format=json


```
{
 "href":"https://demo.wrm247.com/rest/v1/sites/64675?format=json",
 "expandable":"assets.children, assets.dataNodes, …",
 "id":64675,
 "name":"Some Site",
 "parameters":{
    "href":"https://demo.wrm247.com/rest/v1/sites/64675/parameters"
 },
 "assets":{
    "href":"https://demo.wrm247.com/rest/v1/sites/64675/assets"
 },
 "dataNodes":{
    "href":"https://demo.wrm247.com/rest/v1/sites/64675/datanodes"
 },
 "enterprise":{
    "href":"https://demo.wrm247.com/rest/v1/sites/64675/enterprise"
 }
}
```

2.5 Get Assets

URL: /assets
HTTP method: GET

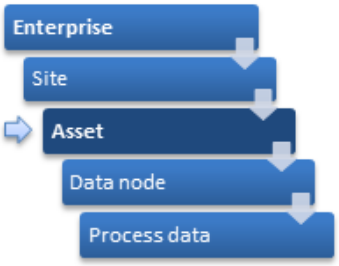


Table 11 - URL Parameters for Get Assets

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/assets?callback=foo returns: foo(response data);
format	Format (json or xml) default is json	/assets?format=xml
parameters	Comma separated array of asset parameter and value pairs surrounded with square brackets.	/assets?parameters=[someParam=true, someOtherParam=124]
enterprises	comma separated enterprise identifiers	/assets?enterprises=WAPICELTD&sites=SITE1,SITE2
sites	comma separated site identifiers	/assets?sites=SITE1,SITE2
parents	comma separated asset parent identifiers	/assets?sites=SITE1&parents=ASSET1
identifiers	comma separated asset identifiers	/assets?sites=SITE1&identifiers=ASSET1,ASSET2
expand	comma separated object field names to expand	/assets?sites=SITE1&identifiers=ASSET1,ASSET2&expand=items.name

Table 12 - Response to Get Assets

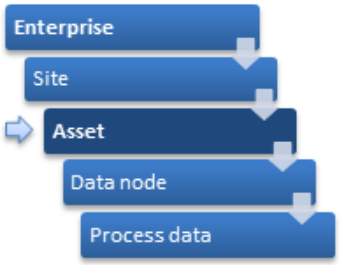
Field	Description
href	URL to this resource
offset	How many results to skip from beginning
limit	Amount of results per page
first	URL to first page
previous	URL to previous page
next	URL to next page
last	URL to last page
items	Array of href elements containing URLs to the returned assets

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/assets?format=json


```
{
 "href":"https://demo.wrm247.com/rest/v1/assets?format=json",
 "expandable":"children,dataNodes,enterprise,id,model, …",
 "offset":0,
 "limit":10,
 "next":"https://demo.wrm247.com/rest/v1/assets?format=json&offset=10",
 "last":"https://demo.wrm247.com/rest/v1/assets?format=json&offset=30",
 "fullSize":32,
 "items":[
           {"href":"https://demo.wrm247.com/rest/v1/assets/3711"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/169"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/205"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/48"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/51"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/121"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/64678"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/74493"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/11"},
           {"href":"https://demo.wrm247.com/rest/v1/assets/69466"}
  ]
}
```

2.6 Asset Object



URL: /assets/123 *(where 123 is globally unique id for the asset)*
HTTP method: GET

Table 13 - URL Parameters for Asset Object

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/assets/123?callback=foo returns: foo: foo(response data);
format	Format (json or xml) default is json	/assets/123?format=xml
expand	Comma separated field names to expand	/assets/123?expand=asset, functionality

Table 14 - Response to Asset Object

Field	Description
href	URL to this resource
name	User friendly name of the asset
identifier	Unique identifier for an Asset in Site context. This can be used for query filtering when searching for assets or data nodes.
model	Asset model name
parameters	Parameters contain meta-data of the asset. Objects containing parameters and their values inside an array: *[* *{"parameter":"example1","value":"exampleValue1"},* *{"parameter":"example2","value":"exampleValue2"}* *]*
dataNodes	Object containing URL to the data nodes that belong to this asset
children	Object containing URL to the child assets that belong to this asset
parent	Object containing URL to the parent asset
site	Object containing URL to the site this asset belongs to
enterprise	Object containing URL to the enterprise this asset belongs to

Table 15 - Asset Object Sub-URLs

URL	Description	Supported query parameters
/assets/{assetId}/children	list assets under the specified asset	name callback format expand offset limit
/assets/{assetId}/datanodes	lists datanodes under the specified asset	name callback format expand offset limit

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/assets/3711?format=json


```
{
 "href":"https://demo.wrm247.com/rest/v1/assets/3711?format=json",
 "expandable":"children.children, children.dataNodes, …"
 "id":3711,
 "name":"Some Asset",
 "model":"Generic Asset",
 "parameters":{
    "href":"https://demo.wrm247.com/rest/v1/assets/3711/parameters"
 },
 "dataNodes":{
    "href":"https://demo.wrm247.com/rest/v1/assets/3711/datanodes"
 },
 "children":{
    "href":"https://demo.wrm247.com/rest/v1/assets/3711/children"
 },
 "parent":{
    "href":"https://demo.wrm247.com/rest/v1/assets/3711/parent"
 },
 "site":{
    "href":"https://demo.wrm247.com/rest/v1/assets/3711/site"
 },
 "enterprise":{
    "href":"https://demo.wrm247.com/rest/v1/assets/3711/enterprise"
 }
}
```

2.7 Get Data Nodes

URL: /datanodes
HTTP method: GET

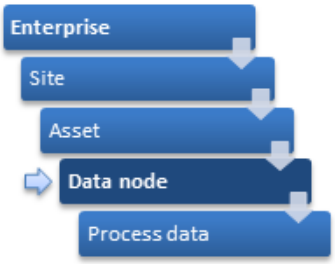


Table 16 - URL Parameters for Get Data Nodes

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/datanodes?callback=foo returns: foo: foo(response data);
format	Format (json or xml) default is json	/datanodes?format=xml
functionalities	comma separated functionality names	/datanodes?functionalities=ENGINE_RPM,ENGINE_TEMP
functionalityParameters	Comma separated array of functionality parameter and value pairs surrounded with square brackets.	/datanodes?functionalities=ENGINE_RPM&parameters=[someParam=true,someOtherParam=124]
enterprises	comma separated enterprise identifiers	/datanodes?enterprises=WAPICELTD&sites=SITE1,SITE2
sites	comma separated site identifiers	/datanodes?functionalities=ENGINE_RPM,ENGINE_TEMP&sites=SITE1,SITE2
assets	comma separated asset identifiers	/datanodes?functionalities=ENGINE_RPM,ENGINE_TEMP&sites=SITE1,SITE2&assets=ASSET1,ASSET2
identifiers	comma separated data node identifiers	/datanodes?sites=SITE1&assets=ASSET1,ASSET2&identifiers=DN1,DN2
expand	comma separated object field names to expand	/datanodes?sites=SITE1&assets=ASSET1,ASSET2&expand=items.name

Table 17 - Response to Get Data Nodes

Field	Description
href	URL to this resource
offset	How many results to skip from beginning
limit	Amount of results per page
first	URL to first page
previous	URL to previous page
next	URL to next page
last	URL to last page
items	Array of href elements containing URLs to the returned data nodes

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/datanodes?format=json


```
{
 "href":"https://demo.wrm247.com/rest/v1/datanodes?format=json",
 "expandable":"asset,enterprise,functionality,id,model,name, …",
 "offset":0,
 "limit":10,
 "next":"https://demo.wrm247.com/rest/v1/datanodes?format=json&offset=10",
 "last":"https://demo.wrm247.com/rest/v1/datanodes?format=json&offset=140",
 "fullSize":144,
 "items":[
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/155"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/157"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/158"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/159"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/160"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/161"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/17130"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/17222"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/15839"},
           {"href":"https://demo.wrm247.com/rest/v1/datanodes/50287"}
 ]
}
```

2.8 Data Node Object

URL: /datanodes/123 *(where 123 is globally unique id for the data node)*
HTTP method: GET

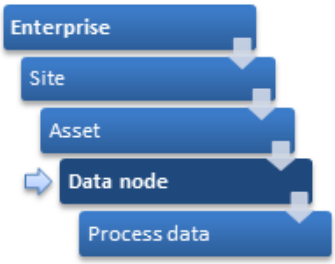


Table 18 - URL Parameters for Data Node Object

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/datanodes/123?callback=foo returns: foo: foo(response data);
format	Format (json or xml) default is json	/datanodes/123?format=xml
expand	Comma separated field names to expand	/datanodes/123?expand=asset, functionality

Table 19 - Response to Data Node Object

Field	Description
href	URL to this resource
name	User friendly name of the data node
unit	Unit for process data
identifier	Unique identifier for Data Node in Asset context. This can be used for query filtering when searching for data nodes.
writable	If "true" this data node can be written to. See chapter 2.10 for more info.
readable	If "true" this data node can be read from. See chapter 0 for more info.
online	If "true" the device that this data node is configured to is online and resources defined in chapters 2.10 and 0 can be used.
functionality	Functionality Object containing URL to the functionality
functionalityParameters	Objects containing functionality parameters and their values inside an array. These parameters can be used to further identify correct data node if for example one asset contains multiple data nodes with same functionality names.
processData	Object containing link to process data for this data node
asset	Asset Object containing URL to the asset
site	Site Object containing URL to the site this data node belongs to
enterprise	Object containing URL to the enterprise this data node belongs to

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/datanodes/16960?format=json


```
{
 "href":"https://demo.wrm247.com/rest/v1/datanodes/16960?format=json",
 "expandable":"asset.children,asset.dataNodes,asset.enterprise,asset.id, …",
 "id":16960,
 "name":"Acc z",
 "model":"Generic DataNode",
 "parameters":{
    "href":"https://demo.wrm247.com/rest/v1/datanodes/16960/parameters"
 },
 "processData":{
    "href":"https://demo.wrm247.com/rest/v1/datanodes/16960/processdata"
 },
 "asset":{
    "href":"https://demo.wrm247.com/rest/v1/datanodes/16960/asset"
 },
 "site":{
    "href":"https://demo.wrm247.com/rest/v1/datanodes/16960/site"
 },
 "enterprise":{
    "href":"https://demo.wrm247.com/rest/v1/datanodes/16960/enterprise"
 }
}
```

2.9 Get Process Data

URL: /datanodes/123/processdata *(where 123 is globally unique id for the data node)*
HTTP method: GET

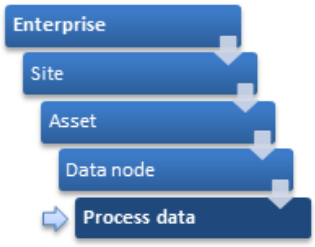


Table 20 - URL Parameters for Get Process Data

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/datanodes/123/processdata?callback=foo returns: foo: foo(response data);
format	Format (json or xml) default is json	/datanodes/123/processdata?format=xml
expand	Comma separated field names to expand	/datanodes/123/processdata?expand=dataNode
begin	Unix time in **microseconds** (UTC). Measurements are queried starting from this time. If end parameter is not given, will return values starting from the begin time until limit is reached. **NOTE: If no begin parameter is given, only latest value is returned.**	/datanodes/123/processdata?begin=1346842881000
end	Unix time in **microseconds** (UTC). Measurements are queried ending to this time. **NOTE: If using end parameter, begin parameter must be given also.**	/datanodes/123/processdata?begin=1346842881000&end=1346842981000
limit	Maximum amount of measurements to return. Default limit value is 10000 and maximum is 100000. **NOTE: If no begin parameter is given, only latest value is returned.**	/datanodes/123/processdata?begin=1346842881000&limit=10
order	Order the returned process data values by timestamps. Possible values are *ascending* and *descending*. **Default is ascending.**	/datanodes/123/processdata?begin=1346842881000&limit=10&order=descending

Table 21 - Response to Get Process Data

Field	Description
href	URL to this resource
limit	Limit value
begin	Begin time value
end	End time value
next	URL to next page of this query
items	Objects containing timestamp (UTC Unix time in **microseconds**) and process data value inside an array, see Table 22 – Process data value data formats for representation. Field named "ts" is for timestamp and field "v" for value.
dataNode	Object containing URL to the data node that these measurements belong to

Table 22 – Process data value data formats

Data Type	Format
U8,S8,U16,S16,U32, S32,U64,S64,Double	Textual representation of the numerical value, e.g. "-4". For Double data type the decimal mark used is dot. Exponential notation is supported for Double data type.
String	Any Unicode character except " or \ which must be escaped by \
Boolean	"true" or "false" without quotes
Binary	Base64 encoded

Example JSON responses:

Getting latest value:

URL: https://demo.wrm247.com/rest/v1/datanodes/123/processdata

```
{
"href":”https://demo.wrm247.com/rest/v1/datanodes/123/processdata”,
"limit":1,
"begin":,
"end":,
"next":,
"items":[
  {“ts”:132512512511525,”v”:14 }
]
}
```

Getting values in time range:

URL:
https://demo.wrm247.com/rest/v1/datanodes/123/processdata?&begin=12000&end=1242192184&limit=100000

```
{
"href":”https://demo.wrm247.com/rest/v1/datanodes/123/processdata?&begin=12000&end=1242192184&limit=100000”,
"limit":100000,
"begin":12000,
"end":1242192184,
"next":”https://demo.wrm247.com/rest/v1/datanodes/123/processdata?datanode=1&begin=350000&end=1242192184&limit=100000”,
"items":[
  {“ts”:12001,”v”:14 },
  {“ts”:12002,”v”:13 },
  {“ts”:12003,”v”:12 },
… (continues for 100000 values)
]
}
```

2.10 Write Data Node Value

URL: /datanodes/write
HTTP method: POST

This is resource is used to write to data nodes while the server has connection to devices the data nodes are configured to. The connection status for data nodes can be viewed from the data node field "online". Data Node must also be writable, which is specified by data node field "writable".

Writing to a data node is done by using a HTTP POST method and the data enclosed in the request must contain following fields.

Table 23 - POST data for Write Data Node

Field	Description
data	Array of objects containing fields "id" and "v"
id	ID of the data node that is to be written
v	Value to be written to data node. Refer to Table 22 – Process data value data formats for format.

Table 24 - Response to Write Data Node

Field	Description
href	URL to this resource.
results	Array of objects containing fields "id","status" and "errorDescription".
id	ID of the data node that was targeted by this read operation.
status	Status of the write operation. See Table 25 - General purpose status codes.
errorDescription	More detailed description of the error. This field is not returned if the operation completed successfully.

Table 25 - General purpose status codes

Code	Description
0	Operation completed successfully.
1	No access. Client does not access to the requested resource or operation.
2	Not available. In case of write/read data node value this is used to indicate that the device where the data node is configured is not online.
3	Invalid input. The data received from client is not formatted correctly, or is not within limits.
4	Internal server error.

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/datanodes/write

POST data in JSON:

```
[
{"id":123,"v":-4},
{"id":124,"v":152},
{"id":125,"v":"TextData"}
]
```

POST data in XML:

```
<writeDataNodeDataList>
  <writeDataNodeData>
    <id>123</id>
    <v>-4</v>
  </writeDataNodeData>
  <writeDataNodeData>
    <id>124</id>
    <v>152</v>
  </writeDataNodeData>
  <writeDataNodeData>
    <id>125</id>
    <v>TextData</v>
  </writeDataNodeData>
</writeDataNodeDataList>
```

Response:

```
{
"href":" https://demo.wrm247.com/rest/v1/datanodes/write"
"results":[
{"id":123, "status":0},
{"id":124, "status":0},
{"id":125, "status":2, "errorDescription":"Data Node is not online"}]
}
```

2.11 Read Data Node Value

URL: /datanodes/read
HTTP method: GET

This is resource is used to read data from data nodes while the server has a connection to devices the data node is configured to. The connection status for data nodes can be viewed from the data node field "online". Data Node must also be readable, which is specified by data node field "readable".

Table 26 - URL Parameters for Read Data Node Value

Parameter	Description	Example
callback	A JavaScript function to run when the response is received (JSONP)	/datanodes/read?dataNodes=123&callback=foo returns: foo: foo(response data);
format	Format (json or xml) default is json	/datanodes/read?dataNodes=123&format=xml
dataNodes	Comma separated data node ids to read. **This parameter is required.**	/datanodes/read?dataNodes=123,124

Table 27 - Response to Read Data Node Value

Field	Description
href	URL to this resource
results	Array of objects containing fields "id", "status", "ts", "v" and "errorDescription"
id	ID of the data node that was targeted by this read operation
status	Status of this read operation. See Table 25 - General purpose status codes.
ts	Timestamp (UTC Unix time in **microseconds**) indicating when this value was read. This field is not returned if the operation did not complete successfully.
v	Value read from data node. Refer to Table 22 – Process data value data formats for format. This field is not returned if the operation did not complete successfully.
errorDescription	More detailed description of the error. This is not returned if the operation completed successfully.

Example JSON response:

URL: https://demo.wrm247.com/rest/v1/datanodes/read?dataNodes=123,124,125


```
{
"href":"https://demo.wrm247.com/rest/v1/datanodes/read?dataNodes=123,124,125",
"results":[
 {
  "id":123,
  "status":0,
  "ts":132512512511525,
  "v":-2
 },
 {
  "id":124,
  "status":0,
  "ts":132512512511525,
  "v":-3
 },
 {
  "id":125,
  "status":2,
  "errorDescription":"Data Node is not online"
 }
]}
```