

DOCUMENT

VOSpace REST API User's Manual

Prepared by Nieto Rodríguez, Sara
Reference
Issue 1
Revision 0
Date of Issue 10-11-2016
Status Issued
Document Type UM
Distribution

APPROVAL

Title VOSpace REST API User Manual	
Issue 1	Revision 0
Author Nieto Rodríguez, Sara	Date 10-11-2016
Approved by	Date

CHANGE LOG

Reason for change	Issue	Revision	Date

CHANGE RECORD

Issue	Revision		
Reason for change	Date	Pages	Paragraph(s)

Table of contents:

1	INTRODUCTION AND SCOPE	4
2	AUTHENTICATION AND AUTHORIZATION	5
3	SERVICE USER'S MANUAL.....	6
3.1	Create new node	6
3.1.1	HTTP Request	6
3.1.2	HTTP Param.....	6
3.1.3	HTTP Input	7
3.1.4	HTTP Response.....	8
3.2	Remove node	8
3.2.1	HTTP Request	8
3.2.2	HTTP Param.....	9
3.2.3	HTTP Response.....	9
3.3	Move and copy node	9
3.3.1	HTTP Request	9
3.3.2	HTTP Param.....	9
3.3.3	HTTP Input	10
3.3.4	HTTP Response.....	10
3.4	Get node.....	11
3.4.1	HTTP Request	11
3.4.2	HTTP Param.....	11
3.4.3	HTTP Response.....	11
3.5	Get nodes shared	12
3.5.1	HTTP Request	12
3.5.2	HTTP Param.....	12
3.5.3	HTTP Response.....	12
3.6	Set node.....	13
3.6.1	HTTP Request	13
3.6.2	HTTP Param.....	13
3.6.3	HTTP Input	14
3.7	Get protocols.....	14
3.7.1	HTTP Request	14
3.7.2	HTTP Param.....	14
3.8	Get views	15
3.8.1	HTTP Request	15
3.8.2	HTTP Param.....	15
3.9	Get properties	16
3.9.1	HTTP Request	16
3.9.2	HTTP Param.....	16
3.10	Push to Vospace	17
3.10.1	HTTP Request	17
3.10.2	HTTP Param.....	17
3.10.3	HTTP Input	17
3.10.4	HTTP Response.....	18
3.11	Pull from Vospace.....	19
3.11.1	HTTP Request	19
3.11.2	HTTP Param.....	19
3.11.3	HTTP input.....	19
4	OTHER FUNCTIONALITIES.....	20

1 INTRODUCTION AND SCOPE

VOSpace is the IVOA protocol for distributed data storage (REC. 29 March 2013). It defines a standard interface, which specifies how VO agents and applications can use network attached data storage to persist and exchange data in a standard way.

The protocol defines a RESTful interface that allows users to store and retrieve data objects over the network.

The **VOSpace Service** implemented by the ESDC at ESAC, for the scientific archives, is based on the version 2.0 of the protocol. The standard way to access to this service is through a Web Client Interface at <https://vospace.esac.esa.int>.

To allow the access to external applications and expert or normal users that need to manage data directly bypassing the Web Interface, an integrated entry point has been developed for VOSpace. This service can be accessed using standard HTTP requests.

The scope of this document is to provide a detailed user manual of the VOSpace RESTful API implemented, describing the usage of the functionalities available, its inputs and outputs.

The following functionalities that will be implemented in future releases are out of the scope of this document:

- Find nodes: searching nodes within the service
- Pull to VOSpace (server-to-server transfer): importing data into a remote VOSpace service
- Push from VOSpace (server-to-server transfer): sending data from a service to another

2 AUTHENTICATION AND AUTHORIZATION

This service provides users with an account in the VOSpace repository. To get a VOSpace account each user must be previously registered in the ESA LDAP server. Then, the LDAP credentials can be used to access VOSpace service that for the first access creates automatically the new account in a seamless way to the user.

The user accounts have a quota limit of 50GB but it would be probably increased in the future according to special users needs.

From the RESTful API point of view, the user should authenticate on every request against the service. This VOSpace API is mainly intended to be used for scripting purposes and it is compliant with the VOSpace specification v2.0. All http request require secure connection through https, otherwise, the response will redirect to the https endpoint.

Regarding the operations allowed in terms of accessing other user accounts within the service are described in the corresponding functions of the following User Manual.

3 SERVICE USER'S MANUAL

The following section describes how to use the different functionalities provided by the service including request, output, parameter and examples of usage.

3.1 Create new node

Create a new node within the service.

3.1.1 HTTP Request

HTTP METHOD	URL
HTTP-PUT	https://<host>/<service>/servlet/nodes/<uname>/<path-of-node>

Example of request:

```
curl -v -u <uname> -X PUT -d @<file-path>
https://<host>/<service>/servlet/nodes/<uname>/<path-of-node>
```

3.1.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name
<path-of-node>	Yes	Path of node to be created

3.1.3 HTTP Input

This functionality requires as input the representation of the Node to be created. This representation should be compliant with the schema v2.0 corresponding to the service specification (<http://www.ivoa.net/xml/VOSpace/v2.0>).

```
<?xml version="1.0" encoding="UTF-8"?>
<vos:node xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="vos:LinkNode"
uri="vos://esavo!vospace/mylink">
  <vos:properties>
    <vos:property uri="vos://esavo!vospace/properties
#date">xxx</vos:property>
    <vos:property uri="vos://esavo!vospace/properties
#description">xxx</vos:property>
    <vos:property uri="vos://esavo!vospace/properties # length "
>xxx</vos:property>
    <vos:property uri="vos://esavo!vospace/properties #
mtime">xxx</vos:property>
    <vos:property uri="vos://esavo!vospace/properties # btime">xxx
  </vos:property>
</vos:properties>
<vos:target>vos://esavo!vospace/mytable</vos:target>
</vos:node>
```

The list of properties accepted by the service is in section 2.9. The length property is automatically created by the service after checking the real size of the file to be uploaded (see section 2.10).

3.1.4 HTTP Response

As response the service will provide the HTTP code response and the representation of the node created with its properties.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<vos:node xmlns:vos=http://www.ivoa.net/xml/VOSpace/v2.0
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="vos:ContainerNode" uri="vos://esavo!vospace/snieto/ContainerTest2">
<vos:properties>
  <vos:property uri="vos://esavo!vospace/properties#date" readOnly="false">29-
07-2014</vos:property>
  <vos:property uri="vos://esavo!vospace/properties#description"
readOnly="false">Second description of the node.</vos:property>
</vos:properties>
<vos:nodes />
</vos:node>
```

In case of failure the service will provide the corresponding HTTP error response.

3.2 Remove node

This operation removes nodes within the space. It's important to remark that this operation does not remove elements from remote services.

In case of nodes of type unstructured, structured and links, this operation will remove the single node requested. In case of container, this operation will also remove all its children.

3.2.1 HTTP Request

HTTP METHOD	URL
HTTP-DELETE	<a href="http://<host>/<service >/servlet/nodes/<uname>/<path-of-node>">http://<host>/<service >/servlet/nodes/<uname>/<path-of-node>

Example of request:

```
curl -v -u <uname> -X DELETE
http://<host>/<service>/servlet/nodes/<uname>/<path-of-node>
```


3.2.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name
<path-of-node>	Yes	Path of node to be removed

3.2.3 HTTP Response

In case of failure the service will provide the corresponding HTTP error response.

3.3 Move and copy node

Move/copy nodes asynchronously within the service. HTTP is the only transfer protocol accepted by this implementation. The UWS protocol (Universal Worker Service), manages the asynchronous job requests through a jobs queue called *async*.

Service operation rules:

- To **move** nodes from other user account within the service is **not** allowed.
- To **copy** nodes from other user account within the service depends on the access granted on the target node.

3.3.1 HTTP Request

HTTP METHOD	URL
HTTP-POST	<a href="http://<host>/<service >/servlet/transfers/<uws-job-list>">http://<host>/<service >/servlet/transfers/<uws-job-list>

Example of request:

```
curl -v -u <uname> -X POST -d @<xml-file-path> http://<host>/<service >/servlet/transfers/async
```

3.3.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name
<uws-job-list>	Yes	async

3.3.3 HTTP Input

Example of file content with transfer details:

```
<vos:transfer xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0" >  
  <vos:target><vospace-node-uri></vos:target>  
  <vos:direction><vospace-node-uri></vos:direction>  
  <vos:keepBytes><copy-move></vos:keepBytes>  
</vos:transfer>
```

To **copy**, the parameter `<copy-move>` will be **true**.

To **move**, the parameter `<copy-move>` will be **false**.

3.3.4 HTTP Response

The request will provide as response the job ID corresponding to the job created by the service to address the transference. Then, the job status can be pooled to the following HTTP address:

```
http://<host>/<service>/servlet/transfers/<uws-job-list>/<job-id>
```

This request will provide the details of the job status: completed, error, pending, etc. Once the job has finished, it should be manually removed from the service. This behavior corresponds to the UWS protocol.

Finally, you can remove jobs from the queue through the following command:

```
curl -v -u <uname> -X DELETE http://<host>/<service>/servlet/transfers/<uws-job-list>/<uws-job-id>
```

In case of failure the service will provide the corresponding HTTP error response.

3.4 Get node

Get the details for a specific Node within the service. In case of containers, the system will retrieve also its children and properties. This operation also allows accessing shared nodes belonging to other users within the service.

3.4.1 HTTP Request

HTTP METHOD	URL
HTTP-GET	<a href="https://<host>/<service >/servlet/nodes/<uname>/<path-of-node>">https://<host>/<service >/servlet/nodes/<uname>/<path-of-node>

Example of request:

```
curl -v -u <uname> -X GET https://<host>/<service>/servlet/nodes/<uname>/<path-of-node>
```

3.4.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name
<path-of-node>	Yes	Path of node to be accessed

3.4.3 HTTP Response

The service will retrieve the Node representation for the requested element. Regarding the properties for the requested node, in case of containers, for simplicity, its length will be 4096KB and not the size of its children.

```
<?xml version="1.0" encoding="UTF-8"?>
<vos:node xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="vos:ContainerNode"
uri="vos://esavo!vospace/snieto/ContainerTest/ContainerTest2">
  <vos:properties>
    <vos:property uri="vos://esavo!vospace/properties#btime"
readOnly="false">10/10/2014 17:50:29</vos:property>
    <vos:property uri="vos://esavo!vospace/properties#date"
readOnly="false">29/07/2014 00:00:00</vos:property>
    <vos:property uri="vos://esavo!vospace/properties#length"
readOnly="false">4096</vos:property>
```

```

        <vos:property uri="vos://esavo!vospace/properties#description"
readOnly="false">Second description of the node.</vos:property>
        <vos:property uri="vos://esavo!vospace/properties#mtime"
readOnly="false">10/10/2014 17:50:29</vos:property>
    </vos:properties>
</vos:nodes />
</vos:node>

```

3.5 Get nodes shared

Retrieve the list of nodes shared with a user as such or as part of a group. It also shows information like the owners of nodes and the role (viewer or editor). Note that this operation is out of the scope of the official specification.

3.5.1 HTTP Request

HTTP METHOD	URL
HTTP-GET	<a href="https://<host>/<service>/servlet/share/user">https://<host>/<service>/servlet/share/user
HTTP-GET	<a href="https://<host>/<service>/servlet/share/group">https://<host>/<service>/servlet/share/group

To retrieve the list of nodes shared with **user as such**:

```
curl -v -u <uname> -X GET "https://<host>/<service-name>/servlet/share/user"
```

To retrieve the list of nodes shared with user **as part of group**:

```
curl -v -u <uname> -X GET "https://<host>/<service-name>/servlet/share/group"
```

3.5.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace

3.5.3 HTTP Response

The service will retrieve the Node representation for the requested element.

3.6 Set node

Set the property values for a specific Node. This method can be used to share nodes with other users and/or groups through the following properties: #groupread, #groupwrite, #userread and #userwrite.

According to the specification:

- This method cannot be used to modify the Node type
- This method cannot be used to modify the accepts or provides list of views for the Node
- This method cannot be used to create children of a Container node

In case of sharing, these are the operations allowed for each role:

- Viewer: get and pull from VOspace
- Editor: create, set, remove, copy and push to VOspace

Move operation is restricted to your user account.

3.6.1 HTTP Request

HTTP METHOD	URL
HTTP-GET	<a href="https://<host>/<service>/servlet/nodes/<uname>/<path-of-node>">https://<host>/<service>/servlet/nodes/<uname>/<path-of-node>

Example of request:

```
curl -v -u <uname> -X POST -d @<file-name>
https://<host>/<service>/servlet/nodes/<uname>/<path-of-node>
```

3.6.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name
<path-of-node>	Yes	Path of node to be accessed

3.6.3 HTTP Input

This functionality retrieves as input the node representation with the properties that should be added, removed and/or modified. The operation is the union of the existing values and the new ones.

3.7 Get protocols

Retrieve the list of the transfer Protocols supported by the space service. This implementation service supports HTTP protocol.

3.7.1 HTTP Request

HTTP METHOD	URL
HTTP-GET	<a href="https://<host>/<service>/servlet/protocols">https://<host>/<service>/servlet/protocols

Example of request:

```
curl -v -u <uname> -X GET "https://<host>/<service>/servlet/protocols"
```

3.7.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name

3.8 Get views

Get a list of the Views and data formats supported by the service.

3.8.1 HTTP Request

HTTP METHOD	URL
HTTP-GET	<a href="https://<host>/<service>/servlet/views">https://<host>/<service>/servlet/views

Example of request:

```
curl -v -u <uname> -X GET "https://<host>/<service>/servlet/views"
```

3.8.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name

3.9 Get properties

Get a list of the Properties supported by the service.

List of properties accepted by the service according to the specification is the following:

- "vos://esavo!vospace/properties#**length**"
- "vos://esavo!vospace/properties#**date**"
- "vos://esavo!vospace/properties#**description**"
- "vos://esavo!vospace/properties#**groupread**"
- "vos://esavo!vospace/properties#**groupwrite**"
- "vos://esavo!vospace/properties#**userread**"
- "vos://esavo!vospace/properties#**userwrite**"
- "vos://esavo!vospace/properties#**publicread**"
- "vos://esavo!vospace/properties#**mtime**"
- "vos://esavo!vospace/properties#**btime**"
- "vos://esavo!vospace/properties#**permission**"

3.9.1 HTTP Request

HTTP METHOD	URL
HTTP-GET	https://<host>/<service>/servlet/ properties

Example of request:

```
curl -v -u <uname> -X GET "https://<host>/<service>/servlet/properties"
```

3.9.2 HTTP Param

HTTP PARAMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name

3.10 Push to Vospace

Request a list of URLs to send data to a VOspace node. The UWS protocol (Universal Worker Service), manages the asynchronous job requests through a jobs queue called *async*.

3.10.1 HTTP Request

HTTP METHOD	URL
HTTP-POST	https://<host>/<service>/servlet/transfers/async

Request example:

```
curl -v -u <uname> -X POST -d @<new-job> https://<host>/<service>/servlet/transfers/<uws-job-list>
```

3.10.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name
<uws-job-list>	Yes	async

3.10.3 HTTP Input

The representation of job to be submitted:

```
<vos:transfer xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0">
  <vos:target><target-node></vos:target>
  <vos:direction><transfer-name></vos:direction>
  <vos:view></vos:view>
  <vos:protocol uri=<protocol-uri>/>
</vos:transfer>
```

HTTP PAREMETER	REQUIRED	VALUE
<protocol-uri>	Yes	"vos://esavo!vospace/core#httput"
<target-node>	Yes	Path to get node
<transfer-name>	Yes	Push to Vospace

HTTP is the only protocol accepted by the service for transfers.

3.10.4 HTTP Response

The response will retrieve the ID of the UWS Job created to handle the request. This job can be pooled to check whether the action has finished.

When the job finishes, the system retrieves as output the URL endpoint to send the data to. The following example shows an output example:

```
<vos:transfer xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0">
  <vos:target>vos://esavo!vospace/snieto/ContainerTest/ContainerTest2</vos:target>
  <vos:direction>pullFromVoSpace</vos:direction>
  <vos:view uri="vos://esavo!vospace/core#fits" original="true" />
  <vos:protocol uri="vos://esavo!vospace/core#httpput">
<vos:endpoint>http://vospace.esac.esa.int/vospace/service/data/<user>/1413388239023A</vos:endpoint></vos:protocol>
</vos:transfer>
```

At this point, the system is ready to send data to the URL endpoint retrieved from the step before.

HTTP METHOD	URL
HTTP-POST	<a href="https://<host>/<service>/servlet/data/<uname>/<job-id>">https://<host>/<service>/servlet/data/<uname>/<job-id>

Example of call:

```
curl -v -u <uname> -X POST -F "file=@<path-file >"
https://<host>/<service>/servlet/data/<uname>/<job-id>
```

Once the task has finished, the job can be manually removed from the *async* queue, using the job ID.

```
curl -v -u <uname> -X DELETE https://<host>/<service>/servlet/transfers/async/<job-id>
```

3.11 Pull from Vospace

Request a set of URLs that the client can read data from. The UWS protocol (Universal Worker Service), manages the asynchronous job requests through a jobs queue called *async*.

3.11.1 HTTP Request

HTTP METHOD	URL
HTTP-POST	<a href="https://<host>/<service>/servlet/transfers/<uws-job-list>">https://<host>/<service>/servlet/transfers/<uws-job-list>
HTTP-GET	<a href="https://<host>/<service>/servlet/data/<uname>/<uws-job-id >">https://<host>/<service>/servlet/data/<uname>/<uws-job-id >

Example of request:

```
curl -v -u <uname> -X POST -d @<file-name>
“https://<host>/<service>/servlet/transfers/async”
```

3.11.2 HTTP Param

HTTP PAREMETER	REQUIRED	VALUE
<host>	Yes	vospace.esac.esa.int
<service>	Yes	vospace
<uname>	Yes	LDAP User Name
<uws-job-list>	Yes (POST Req.)	asyn
<uws-job-id>	Yes (GET Req.)	ID of the UWS Job

3.11.3 HTTP input

The representation of the job to be submitted:

```
<vos:transfer xmlns:vos="http://www.ivoa.net/xml/VOSpace/v2.0">
  <vos:target><target-node></vos:target>
  <vos:direction><transfer-name></vos:direction>
  <vos:view<></vos:view>
  <vos:protocol uri=<protocol-uri>/>
</vos:transfer>
```

HTTP is the only protocol accepted by the service for transfers.

HTTP PAREMETER	REQUIRED	VALUE
<protocol-uri>	Yes	“vos://esavo!vospace/core#httput”
<target-node>	Yes	Path to get node
<transfer-name>	Yes	Push to Vospace

4 OTHER FUNCTIONALITIES

Apart from the functionalities described previously, as it was stated early in this document, the following functionalities are out of the scope of this manual:

- Find nodes
- Pull to VOSpace
- Push from VOSpace

Besides, the ability of creating groups of users through the RESTful API is not provided in this version and it should be done through the Web interface.