

Introduction

The Triofox API is provided as a .Net Windows Communication Foundation (WCF) Service and is installed by default with each Triofox deployment.

Connection Endpoint: `https://node1.example.com/namespace/n.svc/`

- Host : node1.example.com this the DNS name of your node that will have API service running. If you have multiple nodes in the same Triofox server farm, all of the worker nodes have the capability to service the API.
- IIS Application Name: "namespace" - this is the literal name of the application that runs the .Net WCF Service
- n.svc : this is the .Net WCF Service end-point

Table of Contents

1 - How to Read the Syntax

.....1.a - Hello World Example

2 - Authentication

.....2.a - Log In

.....2.b - Log Out

3 - Files and Folders Management

.....3.a - Listing Functions

.....3.a.1 - List All Files and Folders

.....3.a.2 - Search for Files and Folders

.....3.b - Folder Management

.....3.b.1 - Get Folder Information

.....3.b.2 - Create a Folder

.....3.b.3 - Move or Copy All Files in a Folder

.....3.b.4 - Delete a Folder

.....3.b.5 - Subscribe to Folder Change Events

.....3.b.6 - Unsubscribe to Folder Change Events

.....3.c - File Management

.....3.c.1 - Get File Information

.....3.c.2 - Move or Copy a File

.....3.c.3 - Delete a File

.....3.c.4 - Rename a File

.....3.c.5 - See Who Has a Lock on a File

.....3.c.6 - Lock a File

.....3.c.7 - Unlock a File

.....3.c.8 - Get Versions List for a File

.....3.c.9 - Restore a File

.....3.c.10 - Mark a File as Favorite

.....3.c.11 - Upload a File

.....3.c.12 - Download a File

4 - Sharing

.....4.a - Share Management

.....4.a.1 - List All Shared Objects

.....4.a.2 - Enable Public Link for a File or Folder

-4.a.3 - Share a File or Folder
-4.a.4 - Add an Access Control List to a Share
-4.a.5 - Delete an Access Control List from a Share
-4.a.6 - Request a File
-4.a.7 - Lock a File that was Shared with a User
-4.a.8 - Unlock a File that was Shared with a User
-4.a.9 - Create Download Session and Link for a File or Folder

5 - Settings Management

-5.a - Organization Functions
 -5.a.1 - Get Organization Information
 -5.a.2 - Update the Administrator's Email Address
 -5.a.3 - Set Up Active Directory
-5.b - Devices
 -5.b.1 - Get All Devices
 -5.b.2 - Approve a Device
-5.c - Group Policies
 -5.c.1 - Get All Group Policies
 -5.c.2 - Set a Group Policy Setting
-5.d - Branding
 -5.d.1 - Get All Branding Settings

6 - User Management

-6.a - User Functions
 -6.a.1 - Get All Users
 -6.a.2 - Find User By Email
 -6.a.3 - Create a Native User
 -6.a.4 - Change a User's First/Last Names
 -6.a.5 - Reset User Password
 -6.a.6 - Change a User's Active Directory UPN (User Principal Name)
 -6.a.7 - Change a User's Email Address
 -6.a.8 - Change a User's Storage Size
 -6.a.9 - Delete a User
 -6.a.10 - Disable a User
 -6.a.11 - Enable a User
 -6.a.12 - Update a User's Storage Configuration
 -6.a.13 - Create a Proxied AD User
-6.b - Group Management
 -6.b.1 - Create New Group
 -6.b.2 - Delete a Group
 -6.b.3 - Add a User to a Group
 -6.b.4 - Removes a User from a Group
 -6.b.5 - Renames a Group
-6.c - Guest Account Functions
 -6.c.1 - Create a Guest User

7 - System Management

-7.a - Diagnostic Functions
 -7.a.1 - Get Node Performance Information
 -7.a.2 - Get Cluster Performance Information
-7.b - Miscellaneous Functions
 -7.b.1 - Check for Desktop Client Updates

=====

1 - How to Read the Syntax

1.a - Hello World Example

To execute API calls, submit an HTTP request using the specified HTTP Verb (in this case "GET"). The first part of the URL is always your domain name followed by the Endpoint URL (in this case, it is "/namespace/n.svc/helloworld/").

Many API calls also require the user to submit data to the server. In most cases, the data type for the request data is XML and the response type is JSON (Javascript Object Notation, but in this case, the response is XML as indicated by the Response-Format).

This API Endpoint can be used as a quick check to see if the API is live. Typing the URL end point in web browser address bar will show the service status directly in browser. If https is used, it can be used to test whether the SSL certificate was installed successfully.

Request:

- Endpoint URL: /namespace/n.svc/helloworld/
- HTTP-Method: GET
- Response Format: application/xml
- Body:

Not applicable

Response:

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">
Hello World @ namespace @http://node1.example.com/namespace/n.svc/helloworld @
aeef8314-f40a-4115-8393-9a04ddb125a0 @ ::1
</string>
```

2 - Authentication

2.a - Login

Generates a cookie string (aka token) that is required to be passed in the header (x-glad-token) of all authenticated API calls.

Request:

- Endpoint URL: /namespace/n.svc/jsonlogin/
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
- Body:

- JsonLogin: XML Object containing the following properties.
- username: This is usually the email of the first user that was created (aka. the administrator)
- password: Plain text password

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonLogin xmlns="http://tempuri.org/">
<username>admin@example.com</username>
<password>test</password>
</JsonLogin>
```

Response:

```
{
  "JsonLoginResult":
  {
    "Success": true,
    "Reason": "",
    "Context": "",
    "Context1": "",
    "Context2": "",
    "CurrentChangeId": 636517091218360650,
    "Cookie":
    "r6ZOxibe6NaI9FStoH35104pbWNzAnstGhJOfx7GToPGCdZLaqN6KaRchb0H81KaX+evzpuQLmi6xuX1iriii5
    6RwzLSXJPwfG43owo9wtmUoeWD",
    "NSEndPoint": "http://node1.example.com/namespace/n.svc/",
    "LSEndPoint": "http://node1.example.com:8080/localstor/n.svc/",
    "RelayServer": null,
    "Hours": 12,
    "User": {
      "AddOn": "",
      "FirstName": "Default",
      "LastName": "Cluster Admin",
      "Email": "admin@example.com",
      "LastVisit": "/Date(1514316972823-0500)/",
      "Created": "/Date(1514298972820-0500)/",
      "AssignedServer": "$$LB$$",
      "UserGuid": "e5022bd4-48d0-4fff-a78a-1f00c80ae61a",
      "IsAdmin": true,
      "IsGuest": false,
      "IsDeleted": false,
      "IsRootAdmin": true,
      "Quota": 0,
      "QuotaUsed": 0,
      "ProductType": "GCD",
      "HasStorage": true,
      "SnapshotBackup": true,
      "MountCloudStorage": false,
      "DenyAcl": "",
      "OwnerRight": "",
      "ReadRight": "",
      "ShareRight": "",
      "WriteRight": "",
      "EnforceQuotaOnClient": false,
      "Edition": "team",
      "ADDomain": "",
      "DeleteRight": "",
    }
  }
}
```

```

"ListRight": "",
"MemberOf": "",
"Description": "",
"Sid": "",
"IsADUser": false,
"UserType": "9379F3BF-16C4-4B24-8FF7-5164142F0A47",
"DisableClientSideCache": false,
"IsBizAccount": false,
"IsPersonalAccount": false,
"InstanceId": "",
"ForceVersioned": false,
"ForceOffline": false,
"EncId": "",
"TeamFolderOwner": false,
"CreateByEmail": ""
},
"ForceVersioned": false,
"PortalServer": "http://localhost/portal/",
"SvrVersion": 224961121,
"SvrVersion1": 224961121,
"ZoneLogin": false,
"BrandingProductName": "CS",
"KeepCmdChannel": false,
"SupportFolderSetLastModified": true,
"SupportSetLastModified": true,
"WCVersion": ""
}
}

```

2.b - Log Out

Expires the current authentication cookie.

Request:

- Endpoint URL: /namespace/n.svc/jsonlogout/
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:

Not applicable

Response:

```

{
"Success": true,
"Reason": "",
"Context": "",
"Context1": "",
"Context2": ""
}

```

=====

3 - Files and Folders Management

3.a - Listing Functions

3.a.1 - List All Files and Folders

Gets a list of folder and file objects containing information and permissions necessary to assemble the file browser tree on the web portal.

Request:

- Endpoint URL: `/namespace/n.svc/jsondir/`
- HTTP-Method: GET
- Response Format: `application/json`
- Headers:
 - `x-glad-token`: [Cookie value from jsonlogin here]
 - `x-glad-altpath`: [relative path to the root. Example: `"documents/"`]
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "SupportWebChunk": true,
  "ViewOnly": false,
  "Path": null,
  "Marker": null,
  "DisableVersionedFolder": false,
  "IsTruncated": false,
  "Contents":
  [
    {
      "LastDeleteTick": 0,
      "LastDeleteUserId": null,
      "Owner": false,
      "SPVendorDataEx": "",
      "IsFolder": true,
      "Key": "My Normal Folder",
      "LastModified": "2017-11-10 15:55:28Z",
      "ETag": "376404065",
      "Size": 0,
      "IconClass": "",
      "Context": null,
      "Version": 0,
      "LastOp": "u",
      "OrigKey": null,

```

```
"Attribute": 0,
"SPID": "",
"AliasLocation": "",
"ResolvedPath": "",
"SPVendorData": "",
"EDTTag": "",
"AccessControl": 3,
"Perm": 120,
"VersionCount": 0,
"LastUpdaterGuid": "00000000-0000-0000-0000-000000000000",
"LastUpdateUserName": "",
"Ext": "",
"ItemVendorData": "",
"Faved": false,
"HasRealVersion": false,
"InplaceOpen": false,
"LocalOnly": false,
"NTFSCanWrite": true,
"SupportCut": true,
"SupportFolderTimeStamp": true,
"SupportMove": true,
"SupportPreview": false,
"LastUpdateSec": 0,
"LastUpload": "2017-11-10 15:55:18Z"
},
{
  "LastDeleteTick": 0,
  "LastDeleteUserId": null,
  "Owner": false,
  "SPVendorDataEx": "",
  "IsFolder": true,
  "Key": "43811DCD-8C5A-4EF9-B903-2EEA9CE88487",
  "LastModified": "2018-01-19 20:03:02Z",
  "ETag": "-1732251141",
  "Size": 0,
  "IconClass": "",
  "Context": null,
  "Version": 0,
  "LastOp": "u",
  "OrigKey": null,
  "Attribute": 0,
  "SPID": "",
  "AliasLocation": "",
  "ResolvedPath": "",
  "SPVendorData": "",
  "EDTTag": "",
  "AccessControl": 3,
  "Perm": 120,
  "VersionCount": 0,
  "LastUpdaterGuid": "00000000-0000-0000-0000-000000000000",
  "LastUpdateUserName": "",
  "Ext": "",
  "ItemVendorData": "",
  "Faved": false,
  "HasRealVersion": false,
  "InplaceOpen": false,
```

```
"LocalOnly": false,
"NTFSCanWrite": true,
"SupportCut": true,
"SupportFolderTimeStamp": true,
"SupportMove": true,
"SupportPreview": false,
"LastUpdateSec": 0,
"LastUpload": "2018-01-19 15:51:19Z"
},
{
  "LastDeleteTick": 0,
  "LastDeleteUserId": null,
  "Owner": false,
  "SPVendorDataEx": "",
  "IsFolder": false,
  "Key": "My_File.jpg",
  "LastModified": "2017-11-13 18:43:27Z",
  "ETag": "53657755",
  "Size": 424199,
  "IconClass": "",
  "Context": null,
  "Version": 0,
  "LastOp": "u",
  "OrigKey": null,
  "Attribute": 0,
  "SPID": "",
  "AliasLocation": "",
  "ResolvedPath": "",
  "SPVendorData": "",
  "EDTTag": "",
  "AccessControl": 3,
  "Perm": 120,
  "VersionCount": 0,
  "LastUpdaterGuid": "00000000-0000-0000-0000-000000000000",
  "LastUpdateUserName": "",
  "Ext": "",
  "ItemVendorData": "",
  "Faved": false,
  "HasRealVersion": false,
  "InplaceOpen": false,
  "LocalOnly": false,
  "NTFSCanWrite": true,
  "SupportCut": true,
  "SupportFolderTimeStamp": true,
  "SupportMove": true,
  "SupportPreview": false,
  "LastUpdateSec": 0,
  "LastUpload": "2018-01-19 20:36:34Z"
},
{
  "LastDeleteTick": 0,
  "LastDeleteUserId": null,
  "Owner": false,
  "SPVendorDataEx": "",
  "IsFolder": true,
  "Key": "My Folder",
```

```
"LastModified": "2017-12-26 12:35:59Z",
"ETag": "",
"Size": 0,
"IconClass": null,
"Context": "<StorageDescriptor><Name>My Folder</Name><Guid>{5F2D946D-EBE5-45ee-8F6E-EC37E9CDFEB5}</Guid><Endpoint>http://localhost:8080/localstor/n.svc/</Endpoint><HostId>{8C1737B4-0778-4594-91CD-C6DD0511C6E5}</HostId><LocalMode>Versioned</LocalMode><VendorData>{8C1737B4-0778-4594-91CD-C6DD0511C6E5}:Versioned:Web:web:\\TeamFolder</VendorData><Loc>Web:TeamFolder</Loc></StorageDescriptor>",
"Version": 0,
"LastOp": "u",
"OrigKey": null,
"Attribute": 6,
"SPID": "5f2d946d-ebe5-45ee-8f6e-ec37e9cdfeb5",
"AliasLocation": "",
"ResolvedPath": "",
"SPVendorData": "{8C1737B4-0778-4594-91CD-C6DD0511C6E5}:Versioned:Web:web:\\TeamFolder",
"EDTTag": "",
"AccessControl": 3,
"Perm": 120,
"VersionCount": 0,
"LastUpdaterGuid": "00000000-0000-0000-0000-000000000000",
"LastUpdateUserName": "",
"Ext": "",
"ItemVendorData": "",
"Faved": false,
"HasRealVersion": false,
"InplaceOpen": false,
"LocalOnly": false,
"NTFSCanWrite": true,
"SupportCut": false,
"SupportFolderTimeStamp": true,
"SupportMove": true,
"SupportPreview": false,
"LastUpdateSec": 0,
"LastUpload": ""
},
{
  "LastDeleteTick": 0,
  "LastDeleteUserId": null,
  "Owner": false,
  "SPVendorDataEx": "",
  "IsFolder": true,
  "Key": "My Versioned Folder",
  "LastModified": "2018-01-17 17:53:04Z",
  "ETag": "",
  "Size": 0,
  "IconClass": null,
  "Context": "<?xml version='1.0' encoding='utf-8'><StorageDescriptor><Name>My Versioned Folder</Name><Guid>5f2d946d-ebe5-45ee-8f6e-ec37e9cdfeb5</Guid><Endpoint>http://localhost:8080/localstor/n.svc/</Endpoint><ServiceType>0</ServiceType><Quota>0</Quota><Expire>9999-12-31 23:59:59Z</Expire><NameAlias /><Bucket /><VendorData>{8C1737B4-0778-4594-91CD-
```

```
C6DD0511C6E5}:Versioned:Web:web:\\TeamFolder</VendorData><Loc>Web:TeamFolder</Loc><LocalMode>Versioned</LocalMode><HostId>{8C1737B4-0778-4594-91CD-C6DD0511C6E5}</HostId><Versioned>false</Versioned><InplaceVersioned>false</InplaceVersioned><IsDir>false</IsDir><DataAtRestEncryption>false</DataAtRestEncryption></StorageDescriptor>",
"Version": 0,
"LastOp": "u",
"OrigKey": null,
"Attribute": 6,
"SPID": "5f2d946d-ebe5-45ee-8f6e-ec37e9cdf5",
"AliasLocation": "",
"ResolvedPath": "",
"SPVendorData": "{8C1737B4-0778-4594-91CD-C6DD0511C6E5}:Versioned:Web:web:\\TeamFolder",
"EDTTag": "",
"AccessControl": 3,
"Perm": 120,
"VersionCount": 0,
"LastUpdaterGuid": "00000000-0000-0000-0000-000000000000",
"LastUpdateUserName": "",
"Ext": "",
"ItemVendorData": "",
"Faved": false,
"HasRealVersion": false,
"InplaceOpen": false,
"LocalOnly": false,
"NTFSCanWrite": true,
"SupportCut": false,
"SupportFolderTimeStamp": true,
"SupportMove": true,
"SupportPreview": false,
"LastUpdateSec": 0,
"LastUpload": ""
},
],
"Quota": 0,
"Used": 151324703,
"CanRead": true,
"CanWrite": true,
"CanWriteFile": true,
"IsGuest": false,
"Perm": 120,
"ChangeVer": 340377093,
"DChangeVer": 348128546,
"HasDeleted": true,
"SupportPurge": true,
"Versioned": false,
"HasSupportPreviewFile": false,
"NTFSCanWrite": true,
"FolderSize": 0,
"Items": 0,
"TotalItems": 0
}
```

3.a.2 - Search for Files and Folders

Gets a list of folder and file objects matching the x-glad-filter header.

Request:

- Endpoint URL: /namespace/n.svc/jsondir/
- HTTP-Method: GET
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-filter: [search string + "^names^25". Example: "screenshot^names^25"]
- Body:

Not applicable

Response:

```
{
  "Success":true,
  "Reason":"",
  "Context":"",
  "Context1":"",
  "Context2":"",
  "SupportWebChunk":true,
  "ViewOnly":false,
  "Path":null,
  "Marker":null,
  "DisableVersionedFolder":false,
  "IsTruncated":false,
  "Contents":
  [
    {
      "LastDeleteTick": 0,
      "LastDeleteUserId": null,
      "Owner":false,
      "SPVendorDataEx":"",
      "IsFolder":false,
      "Key":"my_screenshot.jpg",
      "LastModified":"Monday, November 20, 2017 1:50 PM",
      "ETag":null,
      "Size":406497,
      "IconClass":"",
      "Context":null,
      "Version":0,
      "LastOp":"u",
      "OrigKey":null,
      "Attribute":0,
      "SPID":"",
      "AliasLocation":"screen2.jpg",
      "ResolvedPath":"",
      "SPVendorData":"",
      "EDTTag":"",
      "AccessControl":3,
    }
  ]
}
```

```

"Perm":120,
"VersionCount":0,
"LastUpdaterGuid":"00000000-0000-0000-0000-000000000000",
"LastUpdateUserName":"","
"Ext":"jpg",
"ItemVendorData":"","
"Faved":false,
"HasRealVersion":false,
"InplaceOpen":false,
"LocalOnly":false,
"NTFSCanWrite":true,
"SupportCut":true,
"SupportFolderTimeStamp":false,
"SupportMove":true,
"SupportPreview":true,
"LastUpdateSec":636467826210000000,
"LastUpload":""
}
],
"Quota":0,
"Used":151324703,
"CanRead":true,
"CanWrite":false,
"CanWriteFile":false,
"IsGuest":false,
"Perm":104,
"ChangeVer":340377093,
"DChangeVer":349919671,
"HasDeleted":false,
"SupportPurge":true,
"Versioned":false,
"HasSupportPreviewFile":true,
"NTFSCanWrite":true,
"FolderSize":10233200,
"Items":25,
"TotalItems":0
}

```

3.b - Folder Management

3.b.1 - Get Folder Information

Gets information regarding user quota, access permissions, folder types, etc.

Request:

- Endpoint URL: /namespace/n.svc/getfolderinfo
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]

- x-glad-altpath: [The name of the new folder, relative to the root. Example: "documents/myfolder"]

- Body:

- GetFolderInfo: XML Object containing the following properties.
- Path: Leave empty, pass the folder path in the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetFolderInfo xmlns="http://tempuri.org/">
<Path></Path>
</GetFolderInfo>
```

Response:

```
<GetFolderInfoResponse xmlns="http://tempuri.org/">
<GetFolderInfoResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<Success xmlns="">true</Success>
<Reason xmlns=""/>
<Context xmlns=""/>
<Context1 xmlns=""/>
<Context2 xmlns=""/>
<FolderQuota xmlns="">0</FolderQuota>
<FolderUsed xmlns="">0</FolderUsed>
<CanRead xmlns="">true</CanRead>
<CanShare xmlns="">true</CanShare>
<IsTeamFolder xmlns="">false</IsTeamFolder>
<Subscribed xmlns="">false</Subscribed>
</GetFolderInfoResult>
</GetFolderInfoResponse>
```

3.b.2 - Create a Folder

Gets information regarding user quota, access permissions, folder types, etc.

Request:

- Endpoint URL: /namespace/n.svc/jsondir

- HTTP-Method: PUT

- Response Format: application/json

- Headers:

- x-glad-token: [Cookie value from jsonlogin here]
- x-glad-altpath: [The name of the new folder, relative to the root. Example: "documents/mynewfolder"]

- Body:

Not applicable

Response:

```
{
"Success": true,
"Reason": "",
"Context": "",
"Context1": "",
```

```
"Context2": ""
}
```

3.b.3 - Move or Copy All Files in a Folder

Moves or copies all the files in a folder to a new destination. Both the folder and the destination must exist at the time of the request. If the x-glad-copyonly parameter is set to false, the files will be moved. If the x-glad-copyonly parameter is set to true, the files will be copied.

Request:

- Endpoint URL: /namespace/n.svc/jsonmovefolderfiles/
- HTTP-Method: PUT
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The source folder. Example: "documents/mysourcefolder"]
 - x-glad-newname: [The destination folder. Example: "documents/mydestinationfolder"]
 - x-glad-copyonly: [Boolean. Example: true or false]

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.b.4 - Delete a Folder

Deletes a folder. This command only deletes the end folder so for example, if you delete "documents/myfolder", only "myfolder" gets deleted. The "documents" folder is left intact.

Request:

- Endpoint URL: /namespace/n.svc/jsondir/
- HTTP-Method: DELETE
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The name of the folder that you want to delete, relative to the root. Example: "documents/myfolder"]
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",

```

```
"Context": "",
"Context1": "",
"Context2": ""
}
```

3.b.5 - Subscribe to Folder Change Events

Subscribes the current logged in user to receive email notifications for file-change events.

Request:

- Endpoint URL: /namespace/n.svc/jsonsubscribetofolderchange/
- HTTP-Method: GET
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The name of the folder that you want to subscribe to, relative to the root. Example: "documents/myfolder"]
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.b.6 - Unsubscribe to Folder Change Events

Unsubscribes the current logged in user from email notifications for file-change events.

Request:

- Endpoint URL: /namespace/n.svc/jsonunsubscribefolderchange/
- HTTP-Method: GET
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The name of the folder that you want to subscribe to, relative to the root. Example: "documents/myfolder"]
- Body:

Not applicable

Response:

```
{
  "Success": true,
```

```
"Reason": "",
"Context": "",
"Context1": "",
"Context2": ""
}
```

3.c - File Management

3.c.1 - Get File Information

Gets basic information about a file, such as upload and modification dates, size (in bytes), owner ID, file ID, etc.

Request:

- Endpoint URL: `/namespace/n.svc/jsonfileinfo/`
- HTTP-Method: GET
- Response Format: `application/json`
- Headers:
 - `x-glad-token`: [Cookie value from jsonlogin here]
 - `x-glad-altpath`: [The name of the file relative to the root. Example: `"documents/myFile.txt"`]
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "FileKey": "DE7A72C96C727463057864A8F19ECBE4",
  "FileName": "myFile.txt",
  "FullPath": "documents/myFile.txt",
  "OwnerId": "e5022bd4-48d0-4fff-a78a-1f00c80ae61a",
  "FileExist": true,
  "LastUpload": "1/31/2018 1:55:42 PM",
  "Size": 18,
  "ETag": null,
  "LastModified": "2010-01-10 04:08:00Z",
  "IsFolder": false
}
```

3.c.2 - Move or Copy a File

Moves a file. Please note that files that have been shared or locked will return an error. If the `x-glad-copyonly` parameter is set to false, the file will be moved. If the `x-glad-copyonly` parameter is set to true, the file will be copied.

Request:

- Endpoint URL: `/namespace/n.svc/jsonmove/`

- HTTP-Method: PUT
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The name of the file that you want to copy or move, relative to the root. Example: "documents/myFile.txt"]
 - x-glad-newname: [The destination path, relative to the root. Example: "documents/somefolder/myFile.txt"]
 - x-glad-copyonly: [Boolean. Example: true or false]
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.c.3 - Delete a File

Deletes a file. Please note that files that have been shared or locked will return an error.

Request:

- Endpoint URL: /namespace/n.svc/jsonfile/
- HTTP-Method: DELETE
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The name of the file that you want to delete, relative to the root. Example: "documents/myFile.txt"]
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.c.4 - Rename a File

Renames a file. Please note that the x-glad-altpath header should be relative to the root, such as documents/myFile.txt. On the other hand, the name parameter that goes in the URL should be the file name only, such as myRenamedFile.txt. Files that have been shared or locked will return an error.

Request:

- Endpoint URL: /namespace/n.svc/jsonrename/?name=myRenamedFile.txt

name: [The new file name]

- HTTP-Method: PUT

- Response Format: application/json

- Headers:

- x-glad-token: [Cookie value from jsonlogin here]
- x-glad-altpath: [The name of the file that you want to rename, relative to the root.

Example: "documents/myFile.txt"]

- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.c.5 - See Who Has a Lock on a File

Returns a tab-separated (\t) string containing the username and email, host name or user ID, cluster admin's domain ID, and file path (pre-pended with the cluster administrator's domain ID). The string is returned in the Reason property of a successful response.

Request:

- Endpoint URL: /namespace/n.svc/jsonwhohaslock

- HTTP-Method: POST

- Response Format: application/json

- Headers:

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]
- x-glad-altpath: [The name of the file to get the lock for, relative to the root.

Example: "documents/Budget.docx"]

- Body:

- JsonWhoHasLockFile: XML Object containing the following properties.
- Path: Leave empty, pass the file path in the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonWhoHasLockFile xmlns="http://tempuri.org/">
<Path></Path>
</JsonWhoHasLockFile>
```

Response:

```
{
  "Success": true,
  "Reason": "Example User (exampleuser@example.com) \tweb\t80955bc7-b181-4f84-9179-4f32afe391ad \t80955bc7-b181-4f84-9179-4f32afe391ad:Budget.docx",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.c.6 - Lock a File

Locks a file given the token of a logged in user. When a file is locked, other users cannot modify the file. An XML body is required, for backward compatibility, but the path of the file should be passed in the header. You can simply pass the XML object with no value for the Path property.

Request:

- Endpoint URL: /namespace/n.svc/jsonlockfile
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here. In this case, you want to pass the Cookie of the user who will be locking the file.]
 - x-glad-altpath: [The path to the file, relative to the root. Example: Documents/MyFile.docx]
- Body:
 - JsonDistributedLockFile: XML Object containing the following properties.
 - Path: Leave empty, pass the file path in the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonDistributedLockFile xmlns="http://tempuri.org/">
<Path></Path>
</JsonDistributedLockFile>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.c.7 - Unlock a File

Unlocks a file given the token of a logged in user. When a file is unlocked, it becomes available for editing to other users. An XML body is required, for backward compatibility, but the path of the file should be passed in the header. You can simply pass the XML object with no value for the Path property.

Request:

- Endpoint URL: `/namespace/n.svc/jsonunlockfile`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:
 - Content-Type: `application/xml`
 - x-glad-token: [Cookie value from jsonlogin here. In this case, you want to pass the Cookie of the user who will be unlocking the file.]
 - x-glad-altpath: [The path to the file, relative to the root. Example: `Documents/MyFile.docx`]
- Body:
 - JsonDistributedUnLockFile: XML Object containing the following properties.
 - Path: Leave empty, pass the file path in the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonDistributedUnLockFile xmlns="http://tempuri.org/">
<Path></Path>
</JsonDistributedUnLockFile>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.c.8 - Get Versions List for a File

Gets an XML object containing information about every change that was made to a file.

Request:

- Endpoint URL: `/namespace/n.svc/getverlist`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:
 - Content-Type: `application/xml`
 - x-glad-token: [Cookie value from jsonlogin here.]
 - x-glad-altpath: [The path to the file, relative to the root. Example: `Documents/MyFile.docx`]
- Body:

◦ **GetVersionList**: XML Object containing the following properties. Leave empty, pass the file path in the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetVersionList xmlns="http://tempuri.org/"></GetVersionList>
```

Response:

```
<GetVersionListResponsexmlns="http://tempuri.org/">
<GetVersionListResultxmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<Success xmlns="">true</Success>
<Reason xmlns=""/>
<Context xmlns=""/>
<Context1 xmlns=""/>
<Context2 xmlns=""/>
<VersionList xmlns="">
<FileVersionNode>
<Key>myFile.txt</Key>
<LastModified>2018-01-26 15:37:00Z</LastModified>
<VersionModifyTime>2018-01-26 15:37:10Z</VersionModifyTime>
<ETag>664903220</ETag>
<Size>11</Size>
<Contexti:nil="true"/>
<Version>225811830</Version>
<LastOp>109</LastOp>
<OrigKey>myFile.txt@m#225811830^e5022bd4-48d0-4fff-a78a-1f00c80ae61a</OrigKey>
<EDTTag>225811830</EDTTag>
<UpdateUid>e5022bd4-48d0-4fff-a78a-1f00c80ae61a</UpdateUid>
<UpdateDisplayName>Default Cluster Admin (admin@example.com)</UpdateDisplayName>
</FileVersionNode>
<FileVersionNode>
<Key>myFile.txt</Key>
<LastModified>2018-01-26 15:36:46Z</LastModified>
<VersionModifyTime>2018-01-26 15:36:54Z</VersionModifyTime>
<ETag>1148482987</ETag>
<Size>10</Size>
<Contexti:nil="true"/>
<Version>225811814</Version>
<LastOp>109</LastOp>
<OrigKey>myFile.txt@m#225811814^e5022bd4-48d0-4fff-a78a-1f00c80ae61a</OrigKey>
<EDTTag>225811814</EDTTag>
<UpdateUid>e5022bd4-48d0-4fff-a78a-1f00c80ae61a</UpdateUid>
<UpdateDisplayName>Default Cluster Admin (admin@example.com)</UpdateDisplayName>
</FileVersionNode>
<FileVersionNode>
<Key>myFile.txt</Key>
<LastModified>2018-01-26 15:36:26Z</LastModified>
<VersionModifyTime>2018-01-26 15:36:39Z</VersionModifyTime>
<ETag>-619372383</ETag>
<Size>11</Size>
<Contexti:nil="true"/>
<Version>225811799</Version>
<LastOp>109</LastOp>
<OrigKey>myFile.txt@m#225811799^e5022bd4-48d0-4fff-a78a-1f00c80ae61a</OrigKey>
<EDTTag>225811799</EDTTag>
<UpdateUid>e5022bd4-48d0-4fff-a78a-1f00c80ae61a</UpdateUid>
```

```
<UpdateDisplayName>Default Cluster Admin (admin@example.com)</UpdateDisplayName>
</FileVersionNode>
</VersionList>
<CanWrite xmlns="">true</CanWrite>
</GetVersionListResult>
</GetVersionListResponse>
```

3.c.9 - Restore a File

Restore a file to an earlier version.

Request:

- Endpoint URL: /namespace/n.svc/restorefile
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here.]
 - x-glad-altpath: [The path to the file, relative to the root. Example: MyVersionedFolder/MyFile.docx]
- Body:
 - RestoreFile: XML Object containing the following properties.
 - Path: Leave empty, pass the file path in the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<RestoreFile xmlns="http://tempuri.org/">
<Path></Path>
</RestoreFile>
```

Response:

```
<RestoreFileResponsexmlns="http://tempuri.org/">
<RestoreFileResultxmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<Successxmlns="">true</Success>
<Reasonxmlns=""/>
<Contextxmlns=""/>
<Context1xmlns=""/>
<Context2xmlns=""/>
</RestoreFileResult>
</RestoreFileResponse>
```

3.c.10 - Mark a File as Favorite

Marks a file so it appears in the File Browser list of favorite files.

Request:

- Endpoint URL: /namespace/n.svc/jsonsetfilefav/
- HTTP-Method: POST
- Response Format: application/json
- Headers:

- x-glad-token: [Cookie value from jsonlogin here.]
- x-glad-setfav: [1 or 0 to mark as favorite or not respectively.]
- x-glad-altpath: [The path to the file, relative to the root. Example: MyVersionedFolder/MyFile.docx]

- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

3.c.11 - Upload a File

Uploads a file to the cloud server via the HTTP PUT protocol.

Request:

- Endpoint URL: /storage/proxiedupload.up
- HTTP-Method: PUT
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here.]
 - x-glad-altpath: [The destination path of the file that you want to upload, relative to the root. Example: documents/MyPicture.png]
- Body:

The raw binary file payload.

Response:

You should get a 200-OK response status if the file was uploaded successfully.

3.c.12 - Download a File

Downloads a file from the cloud server via the HTTP GET protocol.

Request:

- Endpoint URL: /storage/download.dn/f=[PATHHERE]
 - f: [the path to the file, relative to the root. Be sure to properly URL encode this querystring variable. Example:
http://localhost/storage/u.svc/download.dn?f=%2fAsia+Characters%MyFile.pdf]
- HTTP-Method: GET
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here.]
- Body:

The raw binary file payload.

Response:

The response is the raw binary file with a 200-OK response status.

=====

4 - Sharing

4.a - Share Management

4.a.1 - List All Shared Objects

Gets a list of folder and file objects containing information such as, creation date, expiration date, and the share global unique identifier (ShareUid) that is necessary for other API calls.

Request:

- Endpoint URL: /namespace/n.svc/enumpeershare/
- HTTP-Method: GET
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:

Not applicable

Response:

```
{
  "PeerEnumPublishedShareResult":{
    "Success": true,
    "Reason": "",
    "Context": "",
    "Context1": "",
    "Context2": "",
    "ShareList":[
      {
        "ShareUid": "5db302b7-0881-4bec-ae09-472fd7f14663",
        "ShareName": "test23423423",
        "OwnerUid": "80955bc7-b181-4f84-9179-4f32afe391ad",
        "PathLocation": "test23423423",
        "IsDir": true,
        "CreateTime": "/Date(1546439295000-0500) /",
        "ExpireTime": "2019-01-02 14:28:15Z",
        "Memo": "",
        "Password": "",
        "IsPublicUrl": false
      }
    ]
  }
}
```

```
}  
}
```

4.a.2 - Enable Public Link for a File or Folder

Creates an internet link for a file or folder that can be used by anyone to access or download the contents without authentication.

Request:

- Endpoint URL: `/namespace/n.svc/enablefilepubliclink/`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:
 - Content-Type: `application/xml`
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - EnableFilePublicLink: XML Object containing the following properties.
 - FileUri: A file path relative to the root. Example: `documents/myFile.txt`

```
<?xml version="1.0" encoding="UTF-8"?>  
<EnableFilePublicLink xmlns="http://tempuri.org/">  
<FileUri>[File path HERE]</FileUri>  
</EnableFilePublicLink>
```

Response:

```
{  
  "Success": true,  
  "Reason": "",  
  "Context": "http://node1.example.com/storage/u.svc/download.dn/  
fid/116671677460880643_18020043014164665155",  
  "Context1": "",  
  "Context2": ""  
}
```

4.a.3 - Share a File or Folder

Shares a file or folder with anyone with an email address.

Request:

- Endpoint URL: `/namespace/n.svc/jsonshare`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:
 - Content-Type: `application/xml`
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:

- **JsonShareFileFolder**: XML Object containing the following properties.
- **Path**: A file or folder path relative to the root. Example: documents/myFile.txt
- **EmailAccts**: A semi-colon separated list of recipient email addresses. Example: user1@example.com;user2@example.com;user3@example.com
- **isFolder**: A boolean indicating whether the item being share is a folder. Example: false (for a file)
- **fileSize**: An integer indicating the size of the item being shared in bytes. Example: 73728
- **Notes**: A text description of what is being shared. This text gets appended to the email that gets sent to the user. Example: Please take a look at this file.
- **CreateUser**: A boolean indicating whether a guest user should be created for the supplied email addresses. Example: false
- **days**: An integer indicating the number of days until the share expires. Example: 7
- **allowUpdate**: A boolean indicating whether a recipient can modify the content. Example: false
- **viewOnly**: A boolean indicating whether a recipient can only view the content. Example: false
- **pwdProtected**: A boolean indicating whether a recipient will be required to log in in order to access the content. Example: false
- **SharePwd**: A plain text password for the share. Example: myPassw0rd123!
- **downloadEmail**: A boolean indicating whether an email will be sent to the sender once the recipient download the content. Example: false
- **uploadEmail**: A boolean indicating whether an email will be sent to the recipient once the sender uploads new content to the shared folder. Example: false
- **changeEmail**: A boolean indicating whether an email will be sent to the sender once the recipient modifies the content on the shared folder. Example: false

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonShareFileFolder xmlns="http://tempuri.org/">
<Path>documents/myFile.txt</Path>
<EmailAccts>user1@example.com;user2@example.com</EmailAccts>
<isFolder>>false</isFolder>
<fileSize>13</fileSize>
<Notes>Please take a look at this file</Notes>
<CreateUser>>false</CreateUser>
<days>7</days>
<allowUpdate>>false</allowUpdate>
<viewOnly>>false</viewOnly>
<pwdProtected>>false</pwdProtected>
<SharePwd></SharePwd>
<downloadEmail>>true</downloadEmail>
<uploadEmail>>true</uploadEmail>
<changeEmail>>true</changeEmail>
</JsonShareFileFolder>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

4.a.4 - Add an Access Control List to a Share

Adds an Access Control List to a shared object.

Request:

- Endpoint URL: /namespace/n.svc/jsonpeeraddshareacl
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - JsonPeerAddShareACL: XML Object containing the following properties.
 - Path: A file or folder path relative to the root. Example: documents/myFile.txt
 - EmailAccts: A semi-colon separated list of recipient email addresses. Example: user1@example.com;user2@example.com;user3@example.com
 - isFolder: A boolean indicating whether the item being share is a folder. Example: false (for a file)
 - fileSize: An integer indicating the size of the item being shared in bytes. Example: 73728
 - Notes: A text description of what is being shared. This text gets appended to the email that gets sent to the user. Example: Please take a look at this file.
 - CreateUser: A boolean indicating whether a guest user should be created for the supplied email addresses. Example: false
 - days: An integer indicating the number of days until the share expires. Example: 7
 - allowUpdate: A boolean indicating whether a recipient can modify the content. Example: false
 - viewOnly: A boolean indicating whether a recipient can only view the content. Example: false
 - pwdProtected: A boolean indicating whether a recipient will be required to log in in order to access the content. Example: false
 - SharePwd: A plain text password for the share. Example: myPassw0rd123!
 - downloadEmail: A boolean indicating whether an email will be sent to the sender once the recipient download the content. Example: false
 - uploadEmail: A boolean indicating whether an email will be sent to the recipient once the sender uploads new content to the shared folder. Example: false
 - changeEmail: A boolean indicating whether an email will be sent to the sender once the recipient modifies the content on the shared folder. Example: false

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonShareFileFolder xmlns="http://tempuri.org/">
<Path>documents/myFile.txt</Path>
<EmailAccts>user1@example.com;user2@example.com</EmailAccts>
<isFolder>>false</isFolder>
<fileSize>13</fileSize>
<Notes>Please take a look at this file</Notes>
<CreateUser>>false</CreateUser>
<days>7</days>
<allowUpdate>>false</allowUpdate>
<viewOnly>>false</viewOnly>
<pwdProtected>>false</pwdProtected>
<SharePwd></SharePwd>
<downloadEmail>>true</downloadEmail>
<uploadEmail>>true</uploadEmail>
<changeEmail>>true</changeEmail>
</JsonShareFileFolder>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

4.a.5 - Delete an Access Control List from a Share

Deletes an Access Control List from a shared object.

Request:

- Endpoint URL: /namespace/n.svc/peerremoveshareacl
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - PeerRemoveShareACL: XML Object containing the following properties.
 - UserEmail: A valid email address. Example: user1@example.com
 - ShareGuid: The global unique identifier (GUID) string of the shared object. This is obtained from a call to the enumpeershare described above. Example: 5db302b7-0881-4bec-ae09-472fd7f14663

```
<?xml version="1.0" encoding="UTF-8"?>
<PeerRemoveShareACL xmlns="http://tempuri.org/">
  <UserEmail>user1@example.com</UserEmail>
  <ShareGuid>5db302b7-0881-4bec-ae09-472fd7f14663</ShareGuid>
</PeerRemoveShareACL>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

4.a.6 - Request a File

Sends an email to one or more recipients with a link to an upload form. Access to the upload form can be protected with a password and an expiration (in days).

Request:

- Endpoint URL: /namespace/n.svc/jsonrequestfileex
- HTTP-Method: POST

- **Response Format:** application/json
- **Headers:**
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- **Body:**
 - JsonRequestFileEx: XML Object containing the following properties.
 - Path: The virtual path to the folder where the file should be uploaded to the cloud root. Example: /documents
 - EmailAccts: A single email address or a semi-colon separated string of email addresses. Example: user1@example.com;user2@example.com
 - days: An integer that specifies the number of days until the expiration of the link. Example: 5
 - Pwd: A password to protect the upload form page. Example: myPass123!

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonRequestFileEx xmlns="http://tempuri.org/">
<Path>/documents</Path>
<EmailAccts>user1@example.com;user2@example.com</EmailAccts>
<days>5</days>
<Pwd>myPass123!</Pwd>
</JsonRequestFileEx>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

4.a.7 - Lock a File that was Shared with a User

Locks a shared file given the token of a logged in user. When a file is locked, other users cannot modify the file.

Request:

- **Endpoint URL:** /namespace/n.svc/jsonlockfileex
- **HTTP-Method:** POST
- **Response Format:** application/json
- **Headers:**
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The virtual path to the file inside the "Files shared with me" root. Example: Files shared with me/MyFolder/MyFile.docx]
- **Body:**
 - JsonDistributedLockFileEx: XML Object containing the following properties.
 - Path: Leave empty and use the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonDistributedLockFileEx xmlns="http://tempuri.org/">
<Path></Path>
```

```
</JsonDistributedLockFileEx>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

4.a.8 - Unlock a File that was Shared with a User

Unlocks a shared file given the token of a logged in user. When a file is unlocked it becomes available for editing to other users.

Request:

- Endpoint URL: /namespace/n.svc/jsonunlockfileex
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-altpath: [The virtual path to the file inside the "Files shared with me" root. Example: Files shared with me/MyFolder/MyFile.docx]
- Body:
 - JsonDistributedUnLockFileEx: XML Object containing the following properties.
 - Path: Leave empty and use the x-glad-altpath header instead.

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonDistributedUnLockFileEx xmlns="http://tempuri.org/">
  <Path></Path>
</JsonDistributedUnLockFileEx>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

4.a.9 - Create Download Session and Link for a File or Folder

Generates a download URL that can be used to download a file via the browser with the current session.

Request:

- Endpoint URL: /namespace/n.svc/jsonsession/
- HTTP-Method: GET

- **Response Format:** application/json

- **Headers:**

- x-glad-token: [Cookie value from jsonlogin here]
- x-glad-altpath: [The path to the file relative to the root. Example: documents/MyFile.docx]

- **Body:**

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "DomainId": "fee6f7ad-5cf6-4b55-9b4f-7400908be0ff",
  "UserId": "e5022bd4-48d0-4fff-a78a-1f00c80ae61a",
  "Para": null,
  "Meta":
  [
    {
      "Name": "x-glad-token",
      "Value":
      "r6ZOXibe6NaI9FStoH35104pbWNzAnstGhJOfx7GToPGCdZLaqN6KaRchb0H81KaX+evzpuQLmi6xuX1iriiiI5
      6RwzLSXJPwfG43owo9wtmUoeWD",
      "NvGuid": "00000000-0000-0000-0000-000000000000",
      "Id": 0,
      "LastAccessTime": "/Date(1516636325768-0500)/"
    }
  ],
  "StatusCode": 200,
  "StatusDescription": null,
  "Url":
  "http://node1.example.com/storage/download.dn?uf=aHR0cCMvbG9jYWxob3N0L3N0b3JhZ2UvZG93bm
  xvYWQuZG4/dWY9YzJOeVpXVnVNaTVxY0djPQ==",
  "DAREPwd": "",
  "DownloadUser": "",
  "EmailFile": "",
  "EmailUser": ""
}
```

5 - Settings Management

5.a - Organization Functions

5.a.1 - Get Organization Information

Gets general information about the organization, including the email and name of the administrator account.

Request:

- Endpoint URL: /namespace/n.svc/gettenant
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - GetTenant: XML Object containing the following properties.
 - TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff

```
<?xml version="1.0" encoding="UTF-8"?>
<GetTenant xmlns="http://tempuri.org/">
<TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
</GetTenant>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "Tenants":
  [
    {
      "DomainId": "fee6f7ad-5cf6-4b55-9b4f-7400908be0ff",
      "AdminId": "e5022bd4-48d0-4fff-a78a-1f00c80ae61a",
      "ResellerId": "",
      "FirstName": "Default",
      "LastName": "Cluster Admin",
      "Email": "admin@example.com",
      "IsActivated": true,
      "IsDeleted": false,
      "Quota": 0,
      "QuotaUsed": 144432060,
      "UserPlan": "3",
      "UserPlanUsed": "0",
      "PaypalProfileId": "",
      "StorageProviderId": "Default",
      "Edition": "team",
      "Expire": "",
      "ResellerName": "",
      "PromoCode": "",
      "LastCheck": "",
      "TeamStatus": "",
      "LastQuotaScan": "",
      "AccessPoint": "",
      "ClusterName": "",
      "CreatedTime": "12/26/2017 9:36:05 AM",
```

```
"CustDNS": "",
"HostedOwner": "",
"HostedStatus": "",
"Organization": "My Organization",
"TrialTimeDays": 30
}
]
}
```

5.a.2 - Update the Administrator's Email Address

Updates the administrator's email address.

Request:

- Endpoint URL: `/namespace/n.svc/updateadminemail`
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - UpdateTenantAdminEmail: XML Object containing the following properties.
 - TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
 - AdminEmail: An email address. Example: user@example.com

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateTenantAdminEmail xmlns="http://tempuri.org/">
<TenantId>200ab08c-4576-4f78-b7df-84468f8291e8</TenantId>
<AdminEmail>user@example.com</AdminEmail>
</UpdateTenantAdminEmail>
```

Response:

```
{
"Success": true,
"Reason": null,
"Context": "",
"Context1": "",
"Context2": ""
}
```

5.a.3 - Set Up Active Directory

Sets the Active Directory settings programmatically.

Request:

- Endpoint URL: `/namespace/n.svc/setadsettings`
- HTTP-Method: POST
- Response Format: application/json
- Headers:

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

• Body:

- SetActiveDirectorySettings: XML Object containing the following properties.
- TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
- EnableActiveDirectory: A boolean string indicating whether or not to enable Active Directory. Example: true
- LdapServer: The name of the Domain Controller machine. Example: DC1
- UserName: The network username to use to access Active Directory. Example: Administrator
- Password: The password in plain text. Example: MyPassword
- FriendlyDomainName: The domain name of your AD server. Example: mydomain.com
- OrganizationalUnit: The name of the AD organizational unit: Example: My OU
- EnableLdaps: A boolean to enable or disable AD access. Example: true
- NestedGroup: A boolean to indicate whether the Organizational Unit is nested or not. Example: false
- ForestRoot: A boolean to indicate whether the Organizational Unit is the forest root or not. Example: false
- EnableHomeDrive: A boolean to enable home drive. Example: false
- NoUserAutoCreation: A boolean to enable auto import of AD users. Example: false

```
<SetActiveDirectorySettings xmlns="http://tempuri.org/">
<TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
<EnableActiveDirectory>true</EnableActiveDirectory>
<LdapServer>domaincontrollercomputernamehere</LdapServer>
<UserName>administrator</UserName>
<Password>mypassword</Password>
<FriendlyDomainName>mydomain.com</FriendlyDomainName>
<OrganizationalUnit></OrganizationalUnit>
<EnableLdaps>true</EnableLdaps>
<NestedGroup>false</NestedGroup>
<ForestRoot>false</ForestRoot>
<EnableHomeDrive>false</EnableHomeDrive>
<NoUserAutoCreation>false</NoUserAutoCreation>
</SetActiveDirectorySettings>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

5.b - Devices

5.b.1 - Get All Devices

Gets a list of all devices (windows clients, server agents, iOS apps, and Android apps).

Request:

- Endpoint URL: /namespace/n.svc/jsongettenantdevices
- HTTP-Method: POST

- **Response Format:** application/json

- **Headers:**

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

- **Body:**

- **JsonGetTenantDevices:** XML Object containing the following properties.
- **TenantId:** The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonGetTenantDevices xmlns="http://tempuri.org/">
<TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
</JsonGetTenantDevices>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "DeviceList":
  [
    {
      "DomainId": "fee6f7ad-5cf6-4b55-9b4f-7400908be0ff",
      "UserId": "e5022bd4-48d0-4fff-a78a-1f00c80ae61a",
      "UserEmail": "admin@example.com",
      "DeviceType": "Windows Client",
      "HostName": "MyComputer",
      "OsVer": "Windows 10 64bit",
      "ClientVer": "9.1.2417.40322",
      "InstanceId": "{4F89C56E-8CAB-4A3E-A9D6-1A064AD26371}",
      "Status": "accepted",
      "LastLogin": "2018-01-26 14:27:02Z",
      "LastReport": "2018-02-01 17:46:11Z",
      "SyncStatus": 3
    }
  ]
}
```

5.b.2 - Approve a Device

Approves a Device, such as a Windows Client, Android App, iOS App, Mac Client, Windows Phone App, or Server Agent.

Request:

- **Endpoint URL:** /namespace/n.svc/approvedevice

- **HTTP-Method:** POST

- **Response Format:** application/json

- **Headers:**

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

• Body:

- ApproveDevice: XML Object containing the following properties.
- InstanceId: The global unique identifier of the device that you want to approve. Example: 4F89C56E-8CAB-4A3E-A9D6-1A064AD26371

```
<?xml version="1.0" encoding="UTF-8"?>
<ApproveDevice xmlns="http://tempuri.org/">
<InstanceId>4F89C56E-8CAB-4A3E-A9D6-1A064AD26371</InstanceId>
</ApproveDevice>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

5.c - Group Policies

5.c.1 - Get All Group Policies

Gets an array of all set Group Policies settings. Settings which have never been set (left on default values) are not returned.

Request:

- Endpoint URL: /namespace/n.svc/jsongettenantpolicy

- HTTP-Method: POST

- Response Format: application/json

• Headers:

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

• Body:

- GetTenant: XML Object containing the following properties.
- TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff

```
<?xml version="1.0" encoding="UTF-8"?>
<GetTenantGroupPolicy xmlns="http://tempuri.org/">
<TenantId>9508c13b-13f7-4dd2-bfae-bc43b978935a</TenantId>
</GetTenantGroupPolicy>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
}
```

```

"Context1": "",
"Context2": "",
"MetaList":
[
{
"Name": "ClusterImpersonation",
"Value": "true",
"NvGuid": "21066829-a909-437c-bf33-8433a4c664a3",
"UserGuid": "00000000-0000-0000-0000-000000000000",
"Id": 0,
"LastAccessTime": "/Date(1517323718153-0500)/"
},
{
"Name": "cluster_info",
"Value": "pu3YbIBh4p74P9eFhsj8Hcgqcq+jJLe4KgDuygF6W4I=",
"NvGuid": "16443150-b74b-448f-ad9c-4602ef8c3757",
"UserGuid": "00000000-0000-0000-0000-000000000000",
"Id": 0,
"LastAccessTime": "/Date(1514288167263-0500)/"
},
{
"Name": "WinSS@2",
"Value": "",
"NvGuid": "5cbc7a81-483a-4d9c-b00a-cf1fc378f8e7",
"UserGuid": "00000000-0000-0000-0000-000000000000",
"Id": 0,
"LastAccessTime": "/Date(1516378564440-0500)/"
},
{
"Name": "WebOnly@4",
"Value": "True",
"NvGuid": "3f3fa36d-f48e-4681-bc59-9efbc20990c5",
"UserGuid": "00000000-0000-0000-0000-000000000000",
"Id": 0,
"LastAccessTime": "/Date(1516817416063-0500)/"
}
],
"Empty": false
}

```

5.c.2 - Set a Group Policy Setting

Sets the value of a specific Group Policy setting. There are many different group policies settings, so if you want to automate the process of setting a specific property that is not being returned in your query to the `jsongettenantpolicy` endpoint (due to it being left as the default value), you can change the setting's value using the web portal, then the setting Name and possible Value will be returned when you query the `jsongettenantpolicy` endpoint again.

Request:

- Endpoint URL: `/namespace/n.svc/jsonsettenantpolicy`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

• Body:

- SetTenantGroupPolicy: XML Object containing the following properties.
- TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
- Name: A string that represents the programmatic name of the group policy setting in the web portal.
- Value: A string representing the value of the group policy property. Depending on the setting, the value can be numeric, boolean, or alphanumeric. Example: true, 1

```
<?xml version="1.0" encoding="UTF-8"?>
<SetTenantGroupPolicy xmlns="http://tempuri.org/">
<TenantId>9508c13b-13f7-4dd2-bfae-bc43b978935a</TenantId>
<Name>ClusterImpersonation</Name>
<Value>true</Value>
</SetTenantGroupPolicy>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

5.d - Branding

5.d.1 - Get All Branding Settings

Gets an array of all set Branding settings. Settings which have never been set (left on default values) are not returned.

Request:

- Endpoint URL: /namespace/n.svc/getbrandingsettings

- HTTP-Method: POST

- Response Format: application/json

• Headers:

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

• Body:

- GetBrandingSettings: XML Object containing the following properties.
- Brid: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff

```
<?xml version="1.0" encoding="UTF-8"?>
<GetBrandingSettings xmlns="http://tempuri.org/">
<Brid>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</Brid>
</GetBrandingSettings>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "MetaList":
  [
    {
      "Name": "Brand_ProductName",
      "Value": "CS",
      "NvGuid": "00000000-0000-0000-0000-000000000000",
      "UserGuid": "00000000-0000-0000-0000-000000000000",
      "Id": 0,
      "LastAccessTime": "/Date(1517347439636-0500)/"
    },
    {
      "Name": "Brand_WebUITheme",
      "Value": "blue",
      "NvGuid": "00000000-0000-0000-0000-000000000000",
      "UserGuid": "00000000-0000-0000-0000-000000000000",
      "Id": 0,
      "LastAccessTime": "/Date(1517347439652-0500)/"
    }
  ],
  "Empty": false
}
```

6 - User Management

6.a - User Functions

6.a.1 - Get All Users

Gets an array of all users.

Request:

- Endpoint URL: /namespace/n.svc/jsonenumuser/[DOMAINID]
- HTTP-Method: GET
- Response Format: application/json
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
 - x-glad-guest: Optional value. Set to 1 to retrieve a list of guest users. Otherwise, regular users will be returned instead.
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "UserList":
  [
    {
      "AddOn": "bhYrcWTZvCX9i5NemJR+tDYRLl9CfeMzUUY0mtRJEEVar/tAtDvPZNQqNULdZdI0",
      "FirstName": "Default",
      "LastName": "Cluster Admin",
      "Email": "admin@example.com",
      "LastVisit": "/Date(1517344224000-0500)/",
      "Created": "/Date(1514298972820-0500)/",
      "AssignedServer": "$$LB$$",
      "UserGuid": "e5022bd4-48d0-4fff-a78a-1f00c80ae61a",
      "IsAdmin": true,
      "IsGuest": false,
      "IsActivated": true,
      "IsDeleted": false,
      "TrialEndTime": "",
      "DomainId": "fee6f7ad-5cf6-4b55-9b4f-7400908be0ff",
      "IsRootAdmin": true,
      "Quota": 0,
      "QuotaUsed": 144432060,
      "ProductType": "GCLOUD",
      "HasStorage": true,
      "SnapshotBackup": true,
      "MountCloudStorage": true,
      "DenyAcl": "",
      "OwnerRight": "",
      "ReadRight": "",
      "ShareRight": "",
      "WriteRight": "",
      "EnforceQuotaOnClient": true,
      "Edition": "",
      "ADDomain": "",
      "DeleteRight": "",
      "ListRight": "",
      "MemberOf": "",
      "Description": "My Organization",
      "Sid": "",
      "IsADUser": false,
      "UserType": "9379F3BF-16C4-4B24-8FF7-5164142F0A47",
      "DisableClientSideCache": false,
      "IsBizAccount": false,
      "IsPersonalAccount": false,
      "InstanceId": "",
      "ForceVersioned": false,
      "ForceOffline": false,
      "EncId": "",
```

```

"TeamFolderOwner": false,
"CreateByEmail": ""
},
{
"AddOn": "bhYrcWTZvCX9i5NemJR+tDYRLl9CfeMzUUY0mtRJEEVar/tAtDvPZNQqNUlDzdI0",
"FirstName": "user",
"LastName": "one",
"Email": "user1@example.com",
"LastVisit": "/Date(1517344185000-0500)/",
"Created": "/Date(1514302533710-0500)/",
"AssignedServer": "$$LB$$",
"UserGuid": "6ffd0fb7-a7e5-4980-89b1-fd04b7f63bfb",
"IsAdmin": false,
"IsGuest": false,
"IsActivated": true,
"IsDeleted": false,
"TrialEndTime": "",
"DomainId": "fee6f7ad-5cf6-4b55-9b4f-7400908be0ff",
"IsRootAdmin": false,
"Quota": 0,
"QuotaUsed": 0,
"ProductType": "GCLLOUD",
"HasStorage": true,
"SnapshotBackup": true,
"MountCloudStorage": true,
"DenyAcl": "",
"OwnerRight": "",
"ReadRight": "",
"ShareRight": "",
"WriteRight": "",
"EnforceQuotaOnClient": true,
"Edition": "",
"ADDomain": "",
>DeleteRight": "",
>ListRight": "",
>MemberOf": "",
>Description": null,
>Sid": "",
>IsADUser": false,
>UserType": "9379F3BF-16C4-4B24-8FF7-5164142F0A47",
>DisableClientSideCache": false,
>IsBizAccount": false,
>IsPersonalAccount": false,
>InstanceId": "",
>ForceVersioned": false,
>ForceOffline": false,
>EncId": "",
>TeamFolderOwner": false,
>CreateByEmail": ""
}
]
}

```

6.a.2 - Find User By Email

Finds a user given an email address and returns the user's properties.

Request:

- Endpoint URL: /namespace/n.svc/jsonqueryuserbyemail
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - JsonQueryUserByEmail: XML Object containing the following properties.
 - Email: An email address. Example: admin@example.com

```
<JsonQueryUserByEmail xmlns="http://tempuri.org/">
<Email>admin@example.com</Email>
</JsonQueryUserByEmail>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "User": {
    "AddOn": "",
    "FirstName": "Default",
    "LastName": "Cluster Admin",
    "Email": "admin@example.com",
    "LastVisit": "/Date(1514316972823-0500)/",
    "Created": "/Date(1514298972820-0500)/",
    "AssignedServer": "$$LB$$",
    "UserGuid": "e5022bd4-48d0-4fff-a78a-1f00c80ae61a",
    "IsAdmin": true,
    "IsGuest": false,
    "IsActivated": true,
    "IsDeleted": false,
    "TrialEndTime": null,
    "DomainId": "fee6f7ad-5cf6-4b55-9b4f-7400908be0ff",
    "IsRootAdmin": true,
    "Quota": 0,
    "QuotaUsed": 0,
    "ProductType": "GCLOUD",
    "HasStorage": true,
    "SnapshotBackup": true,
    "MountCloudStorage": true,
    "DenyAcl": "",
    "OwnerRight": "",
    "ReadRight": "",
    "ShareRight": "",
    "WriteRight": "",
    "EnforceQuotaOnClient": true,
```

```

"Edition": "",
"ADDomain": "",
>DeleteRight": "",
>ListRight": "",
"MemberOf": "",
>Description": "My Organization",
>Sid": "",
"IsADUser": false,
"UserType": "9379F3BF-16C4-4B24-8FF7-5164142F0A47",
"DisableClientSideCache": false,
"IsBizAccount": false,
"IsPersonalAccount": false,
"InstanceId": "",
"ForceVersioned": false,
"ForceOffline": false,
"EncId": "",
"TeamFolderOwner": false,
>CreateByEmail": ""
}
}

```

6.a.3 - Create a Native User

Creates a new local user programmatically.

Request:

- Endpoint URL: `/namespace/n.svc/jsoncreateuser`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:
 - Content-Type: `application/xml`
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - CreateNativeUser: XML Object containing the following properties.
 - DomainId: The domainid Global Unique Identifier for the cluster administrator. Example: `fee6f7ad-5cf6-4b55-9b4f-7400908be0ff`
 - FirstName: A first name. Example: John
 - LastName: A last name. Example: Smith
 - Email: An email address. Example: `admin@example.com`
 - Password: A plain-text password. Example: `MyP@ssw0rd!`
 - CreateDefaultDir: A boolean indicating whether or not to create the Documents and Pictures folders. Example: `true`
 - QuotaGB: A long integer indicating the maximum amount of Gigabytes to allocate to this user. Example: `admin@example.com`
 - ForceVersioned: A boolean indicating whether or not to force all user-created folders to be versioned. Example: `false`
 - EnableOffline: A boolean indicating whether or not to give the user the option to make offline folders. Example: `false`
 - SendWelcomeEmail: A boolean indicating whether or not to send the user a welcome email based on the template specified under the branding settings. Example: `false`

```

<?xml version="1.0" encoding="UTF-8"?>
<CreateNativeUser xmlns="http://tempuri.org/">
  <DomainId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</DomainId>

```

```
<FirstName>John</FirstName>
<LastName>Smith</LastName>
<Email>admin@example.com</Email>
<Password>MyP@ssw0rd!</Password>
<CreateDefaultDir>false</CreateDefaultDir>
<QuotaGB>2</QuotaGB>
<ForceVersioned>false</ForceVersioned>
<EnableOffline>false</EnableOffline>
<SendWelcomEmail>false</SendWelcomEmail>
</CreateNativeUser>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "Guid": "7b371c01-2bbc-4b24-8135-8f8fa6682767",
  "LapseSeconds": 0
}
```

6.a.4 - Change a User's First/Last Names

Changes the first and/or last name for a given User ID (GUID).

Request:

- Endpoint URL: /namespace/n.svc/changeusername
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - ChangeUserName: XML Object containing the following properties.
 - UserId: The Global Unique Identifier for the user. Example: 7b371c01-2bbc-4b24-8135-8f8fa6682767
 - FirstName: A first name. Example: Bob
 - LastName: A last name. Example: Smith

```
<?xml version="1.0" encoding="UTF-8"?>
<ChangeUserName xmlns="http://tempuri.org/">
  <UserId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</UserId>
  <FirstName>Bob</FirstName>
  <LastName>Smith</LastName>
</ChangeUserName>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
```

```
"Context1": "",
"Context2": ""
}
```

6.a.5 - Reset User Password

Resets the password for a given username. This function only works with native users.

Request:

- Endpoint URL: `/namespace/n.svc/jsonresetpwd`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:
 - Content-Type: `application/xml`
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - ResetUserPassword: XML Object containing the following properties.
 - UserName: The login username of the user. This is usually the email. Example: `user@example.com`
 - Password: A new plain-text password. Example: `MyP@ssw0rd!`

```
<?xml version="1.0" encoding="UTF-8"?>
<ResetUserPassword xmlns="http://tempuri.org/">
<UserName>user@example.com</UserName>
<Password>MyP@ssw0rd!</Password>
</ResetUserPassword>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.6 - Change a User's Active Directory UPN (User Principal Name)

Updates the user principal name in the Triofox application if you have changed it in Active Directory.

Request:

- Endpoint URL: `/namespace/n.svc/jsonchangeupn`
- HTTP-Method: POST
- Response Format: `application/json`
- Headers:
 - Content-Type: `application/xml`
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:

- ResetUserPassword: XML Object containing the following properties.
- Username: The global unique identifier of the user. Example: 70a3d6fa-2345-1612-6f84-g8a7f2142a5c
- NewUpn: A User Principal Name. Example: bob@exampledomain.local

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonChangeUserUpn xmlns="http://tempuri.org/">
<UserId>70a3d6fa-2345-1612-6f84-g8a7f2142a5c</UserId>
<NewUpn>bob@exampledomain.local</NewUpn>
</JsonChangeUserUpn>
```

Response:

```
{
  "Success": true,
  "Reason": null,
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.7 - Change a User's Email Address

Changes the email address for a given user.

Request:

- Endpoint URL: /namespace/n.svc/jsonchangeemail
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - JsonChangeUserEmail: XML Object containing the following properties.
 - UserId: The global unique identifier of the user. Example: 48c7d6fa-6646-4614-8f81-f8a7e2544a8b
 - NewEmail: The new email address. Example: bob@example.com

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonChangeUserEmail xmlns="http://tempuri.org/">
<UserId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</UserId>
<NewEmail>johndoe2@example.com</NewEmail>
</JsonChangeUserEmail>
```

Response:

```
{
  "Success": true,
  "Reason": null,
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.8 - Change a User's Storage Size

Sets the storage size limit in Gigabytes for a given User ID (GUID).

Request:

- Endpoint URL: /namespace/n.svc/setuserstorageplangb
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - JsonChangeUserEmail: XML Object containing the following properties.
 - UserId: The global unique identifier of the user. Example: 48c7d6fa-6646-4614-8f81-f8a7e2544a8b
 - GBCount: A long integer indicating the maximum number of Gigabytes to allocate to this user. Example: 1

```
<?xml version="1.0" encoding="UTF-8"?>
<SetUserStoragePlanGB xmlns="http://tempuri.org/">
  <UserId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</UserId>
  <GBCount>1</GBCount>
</SetUserStoragePlanGB>
```

Response:

```
{
  "Success": true,
  "Reason": null,
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.9 - Delete a User

Deletes a user from the database, with an option to keep the user's personal files in the backend storage.

Request:

- Endpoint URL: /namespace/n.svc/jsondeleteuser
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - DeleteUser: XML Object containing the following properties.
 - UserId: The global unique identifier of the user. Example: 48c7d6fa-6646-4614-8f81-f8a7e2544a8b
 - PurgeStoage: A boolean indicating whether or not to purge the user's files. Example: false

```
<?xml version="1.0" encoding="UTF-8"?>
<DeleteUser xmlns="http://tempuri.org/">
<UserId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</UserId>
<PurgeStoage>>false</PurgeStoage>
</DeleteUser>
```

Response:

```
{
  "Success": true,
  "Reason": null,
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.10 - Disable a User

Disables a user so he/she is not able to access the system.

Request:

- Endpoint URL: /namespace/n.svc/jsondisableuser
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - DisableUser: XML Object containing the following properties.
 - UserId: The global unique identifier of the user. Example: 48c7d6fa-6646-4614-8f81-f8a7e2544a8b

```
<?xml version="1.0" encoding="UTF-8"?>
<DisableUser xmlns="http://tempuri.org/">
<UserId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</UserId>
</DisableUser>
```

Response:

```
{
  "Success": true,
  "Reason": null,
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.11 - Enable a User

Reactivates a user who is currently disabled.

Request:

- Endpoint URL: /namespace/n.svc/jsonenableuser

- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - EnableUser: XML Object containing the following properties.
 - UserId: The global unique identifier of the user. Example: 48c7d6fa-6646-4614-8f81-f8a7e2544a8b

```
<?xml version="1.0" encoding="UTF-8"?>
<EnableUser xmlns="http://tempuri.org/">
<UserId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</UserId>
</EnableUser>
```

Response:

```
{
  "Success": true,
  "Reason": null,
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.12 - Update a User's Storage Configuration

Sets the storage configuration for a given User ID (GUID).

Request:

- Endpoint URL: /namespace/n.svc/jsonupdateuserstorageconfig
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - UpdateUserStorageConfig: XML Object containing the following properties.
 - UserId: The global unique identifier of the user. Example: 48c7d6fa-6646-4614-8f81-f8a7e2544a8b
 - StorageConfigure: A StorageDescriptor XML object encoded as a UTF8 string and then converted to Base 64 for safe HTTP transmission. The StorageDescriptor has the following properties:
 - Name: The StorageProviderId returned from a call to gettenant. Example: Default
 - Guid: A predefined global unique identifier of your storage. Use the following values:
 - Home Storage: 41ED6C26-D5F8-424F-A311-BB4FF88140E5
 - Share Folder: 12D56FBD-B7FC-48B9-8B11-6B989BDFB572
 - Peer Share Folder: 69A884C0-9F3B-4736-AF57-CCA7C3CDC1B8
 - Recent Folder: ACBCC443-8F37-42EE-B2F2-7B9F222E4602
 - Search Folder: 4A1ADDAD-93CF-4F43-A7F3-D1036A522665

- Endpoint: The cloud storage component URL for your type of storage. The possible storage types are:
 - Amazon S3: localhost:8080/gladstor/g.svc/
 - Amazon S3 GovCloud: localhost:8080/gladstor/g.svc/
 - Google Cloud Storage: localhost:8080/googlestor/g.svc/
 - OpenStack Swift without Keystone: localhost:8080/gladopens/g.svc/
 - Amazon S3 Clone: localhost:8080/glads3clone/g.svc/
 - Microsoft Azure Blob Storage: localhost:8080/gladazure/g.svc/
 - OpenStack Swift with KeyStone: localhost:8080/gladhp/g.svc/
 - Nirvanix: localhost:8080/gladnirvanix/g.svc/
 - EMC Atmos2: localhost:8080/gladatmos2/g.svc/
 - UNC Path, Local Drive: localhost:8080/gladfilesvr/f.svc/
- ServiceType: An integer representing the service type. The possible service type values are:
 - Main: 9
 - Home Directory: 8
 - Shared Directory: 7
 - Others: 0
- Quota: An integer representing the storage quota in Gigabytes. Supply "0" for unlimited. Example: 1
- Expire: An expiration date if the storage is a share and expires. The date should be formatted in UTC format (yyyy-mm-ddHH:mm:ssZ. Example: 9999-12-31 23:59:59Z
- NameAlias: An alternate friendly-name for your storage.
- AccessKeyId: The Access Key string used for authentication of some cloud storage providers, such as Amazon S3.
- Username: The local username used to impersonate access to the storage.
- ImpersonationMode: If using impersonation to access the storage, set this value to "SessionImpersonate".
- SecretKey: The Secret Key string used for all types of authentication (equivalent to password).
- Bucket: The name of the storage container for some cloud storage provider, such as Amazon S3.
- Folder: A global unique identifier that is used as the name of a sub-folder within the main storage container/bucket.
- AccessPoint: Must be a valid absolute path or CIFS path, such as C:\MyFolder or \\MyComputer\MyFolder
- Versioned: A boolean indicating whether or not to enable versioning for this Tenant (if the storage supports it). Example: false
- InplaceVersioned: A boolean indicating whether or not to enable in-place versioning for the Tenant (if the storage type supports it). Example: false
- IsDir: A boolean indicating whether or not the storage is a Directory. Example: false

```
<?xml version="1.0" encoding="UTF-8"?>  
<UpdateUserStorageConfig xmlns="http://tempuri.org/">  
  <UserId>48c7d6fa-6646-4614-8f81-f8a7e2544a8b</UserId>  
  <StorageConfig>PFN0b3JhZ2VEZXNjcmlwdG9yPjxOYWl1PjQxZWQ2YzI2LWQ1ZjgtNDIOZilhMzExLWJiNGZmODgxNDBlNTwvTmFtZT48R3VpZD40MWVkNmMyNi1kNWY4LTQyNGYtyTMxMS1iYjRmZjg4MTQwZTU8L0dlawQ+PEVuZHbvaw50Pmh0dHA6Ly9sb2NhbgHvc3Q6ODA4MC9nbGFkZmlsZXN2ci9mLnN2Yy88L0VuZHBvaW50PjxtZXJ2aWNlVhlwZT45PC9TZXJ2aWNlVhlwZT48UXVvdGE+MDwvUXVvdGE+PEV4cGlYT45OTk5LTEyLTmxIDIZoJuU5ojU5WjwwRXhwaxjlPjxOYWllQWxpYXMgLz48VXNlc3kbWU+WWRtaW5pc3RyYXRvckBleGFtcGxlImxvY2FsPC9Vc2VyTmFtZT48U2VjcmV0S2V5PnRlc3QxmMjM8L1NlY3JldEtleT48QnVja2V0IC8+PEFfjY2Vzc1BvaW50PlxcdmfdsGVyZGVza3RvcFxzaGFyZW5hbWU8L0FjY2Vzc1BvaW50PjxJbXB1cnNvbmcF0aW9uTW9kZT5TTXNZazW9uswlwzxzb25hdGU8L0ltcGVyc29uYXRpb25Nb2RlPjxwZXJJZaW9uzWq+zMFsc2U8L1Zlcnpb25lZD48SW5wbGFjZVZlcnpb25lZD50cncVlPC9JbnBsYWNlVmVyc2lvbmVkpjxJc0Rpcj5mYWxzZTwvSXNEaXI+PERhdGFbdFJlc3RFbmNyexB0aW9uPmZhbnHNlPC9EYXRhQXRSZXNORW5jcmlwdGlvbj48L1N0b3JhZ2VEZXNjcmlwdG9yPg==</StorageConfig>  
</UpdateUserStorageConfig>
```

The `StorageConfigure` property above holds the following `StorageDescriptor` object encoded as a UTF8 string and then converted to Base 64. If you are using .Net, you could use the following code to convert your `StorageDescriptor` into a flat string for the `StorageConfigure` value.

```
Convert.ToBase64String(UTF8Encoding.UTF8.GetBytes(StorageConfigureXMLString))
```

Unencoded object:

```
<StorageDescriptor> <Name>Default</Name>
<Guid>41ED6C26-D5F8-424F-A311-BB4FF88140E5</Guid>
<Endpoint>http://localhost:8080/gladfilesvr/f.svc/</Endpoint>
<ServiceType>9</ServiceType>
<Quota>0</Quota>
<Expire>9999-12-31 23:59:59Z</Expire> <NameAlias />
<UserName>administrator@cstack.local</UserName>
<SecretKey>55vWcx31tcq69BYyu9Uj7g==</SecretKey> <AccessKeyId>*</AccessKeyId>
<Folder>22c447de-73d5-4d61-a303-5c3c4209eb25</Folder>
<AccessPoint>\\\\examplenode\\sharename</AccessPoint>
<ImpersonationMode>SessionImpersonate</ImpersonationMode> <Versioned>>false</Versioned>
<InplaceVersioned>>true</InplaceVersioned>
<IsDir>>false</IsDir>
</StorageDescriptor>
```

Response:

```
{
  "Success": true,
  "Reason": null,
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.a.13 - Create a Proxied AD User

Imports a user from a remote Active Directory that is connected to the Triofox server via the Server Agent software.

Request:

- **Endpoint URL:** `/namespace/n.svc/jsonimportuserbyupn`
- **HTTP-Method:** POST
- **Response Format:** application/json
- **Headers:**
 - **Content-Type:** application/xml
 - **x-glad-token:** [Cookie value from jsonlogin here]
- **Body:**
 - **JsonAddProxiedUser:** XML Object containing the following properties.
 - **email:** A valid email address. This will also be used as the login username. Example: user@example.com
 - **sid:** A valid user ID from Active Directory. Example: S-1-5-21-2998206365-3271944564-4074028866-1137

```
<?xml version="1.0" encoding="UTF-8"?>
<ImportAdUserByUPN xmlns="http://tempuri.org/">
<UserUPN>bob@exempldomain.local</UserUPN>
<StoragePlan>1</StoragePlan>
<StorageConfig>PFN0b3JhZ2VEZXNjcmlwdG9yPjxOYW1lPjQxZWQ2YzI2LWQ1ZjgtNDI0Zi1hMzExLWJiNGZm
ODgxNDBlNTwvTmFtZT48R3VpZD40MWWVkNmMyNi1kNWY4LTQyNGYtYTMxMSliYjRmZjg4MTQwZTU8L0dlYWQ+PEV
uZHBvaW50Pmh0dHA6Ly9sb2NhbgHvc3Q6ODA4MC9nbGFkZmlsZXN2ci9mLnN2Yy88L0VuZHBvaW50PjxTZXJ2aW
NlVHlwZT45PC9TZXJ2aWNlVHlwZT48UXVvdGE+MDwvUXVvdGE+PEV4cGlyZT45OTk5LTEyLTMxIDIzOjU5OjU5W
jwvRXhwaXJlPjxOYW1lQWxpYXMQgLz48VXNlck5hbWU+YWRtaW5pc3RyYXRvckBleGFtcGxlLmxvY2FsPC9Vc2Vy
```

```
TmFtZT48U2VjcmlV0S2V5PnRlc3QxMjM8L1NlY3JldEtT48QnVja2V0IC8+PEFjY2Vzc1BvaW50PlxcdmFsdGVyZGVza3RvcFxxzaGFyZW5hbWU8L0FjY2Vzc1BvaW50PjxJbXB1cnNvbWV0aW9uTW9kZT5TZXNzaW9uSW1wZXJzb25hdGU8L0l0tGVyc29uYXRpb25Nb2RlPjxWZXJzaW9uZWQ+ZmFsc2U8L1ZlcnNpb25lZD48SW5wbGFjZVZlcnNpb25lZD50cnVlPC9JbnBsYWNlVmVyc2l0bmVkbPjxJc0Rpcj5mYWxzZTwvSXNEaXI+PERhdGFbdFJlc3RFbmNyeXB0aW9uPmZhbHNlPC9EYXRhQXRSZXN0RW5jcnlwdGlvbj48L1N0b3JhZ2VEZXNjcmlwdG9yPg==</StorageConfig>
</ImportAdUserByUPN>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "Guid": "9379e92d-167b-4552-978d-ac47f9595075",
  "LapseSeconds": 0
}
```

6.b - Group Management

6.b.1 - Create New Group

Creates a new group.

Request:

- Endpoint URL: /namespace/n.svc/jsoncreategroup
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - JsonCreateGroup: XML Object containing the following properties.
 - TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
 - GroupName: The name for the new group. Example: HR Department
 - GroupUsers: An optional string formatted as: [user ID],[user email],[first and last name]. Multiple users can be separated by a newline character escaped for XML transmission ().

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonCreateGroup xmlns="http://tempuri.org/">
  <TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
  <GroupName>My New Group</GroupName>
  <GroupUsers>ad21898a-975b-4c0a-bb8b-5e150ac10f56,user1@example.com,John
Doe&#013e5022bd4-48d0-4fff-a78a-1f00c80ae61a,user2@example.com,Bob Smith</GroupUsers>
</JsonCreateGroup>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.b.2 - Delete a Group

Deletes an existing group.

Request:

- Endpoint URL: /namespace/n.svc/jsondeletegroup
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - JsonDeleteGroup: XML Object containing the following properties.
 - TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
 - GroupId: The global unique identifier of the group that you want to delete. Example: 39484fd1-9103-4a0c-8924-ca2c3f0c1b89

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonDeleteGroup xmlns="http://tempuri.org/">
  <TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
  <GroupId>39484fd1-9103-4a0c-8924-ca2c3f0c1b89</GroupId>
</JsonDeleteGroup>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.b.3 - Add a User to a Group

Adds an existing user to an existing group.

Request:

- Endpoint URL: /namespace/n.svc/jsonaddusertogroup
- HTTP-Method: POST

- Response Format: application/json

- Headers:

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

- Body:

- JsonAddUserToGroup: XML Object containing the following properties.
- TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
- GroupId: The global unique identifier of the group. Example: 39484fd1-9103-4a0c-8924-ca2c3f0c1b89
- UserId: The global unique identifier of the user who you want to add to the group. Example: 6ffd0fb7-a7e5-4980-89b1-fd04b7f63bfb

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonAddUserToGroup xmlns="http://tempuri.org/">
<TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
<GroupId>39484fd1-9103-4a0c-8924-ca2c3f0c1b89</GroupId>
<UserId>6ffd0fb7-a7e5-4980-89b1-fd04b7f63bfb</UserId>
</JsonAddUserToGroup>
```

Response:

```
{
"Success": true,
"Reason": "",
"Context": "",
"Context1": "",
"Context2": ""
}
```

6.b.4 - Removes a User from a Group

Removes a User from a Group.

Request:

- Endpoint URL: /namespace/n.svc/jsondeleteuserfromgroup

- HTTP-Method: POST

- Response Format: application/json

- Headers:

- Content-Type: application/xml
- x-glad-token: [Cookie value from jsonlogin here]

- Body:

- JsonDeleteUserFromGroup: XML Object containing the following properties.
- TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
- GroupId: The global unique identifier of the group. Example: 39484fd1-9103-4a0c-8924-ca2c3f0c1b89
- UserId: The global unique identifier of the user who you want to remove from the group. Example: 6ffd0fb7-a7e5-4980-89b1-fd04b7f63bfb

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonDeleteUserFromGroup xmlns="http://tempuri.org/">
```

```
<TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
<GroupId>39484fd1-9103-4a0c-8924-ca2c3f0c1b89</GroupId>
<UserId>6fffd0fb7-a7e5-4980-89b1-fd04b7f63bfb</UserId>
</JsonDeleteUserFromGroup>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.b.5 - Renames a Group

Renames an existing group.

Request:

- Endpoint URL: /namespace/n.svc/jsonsetgroupname
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - JsonSetGroupName: XML Object containing the following properties.
 - TenantId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
 - GroupId: The global unique identifier of the group. Example: 39484fd1-9103-4a0c-8924-ca2c3f0c1b89
 - GroupName: The new group name. Example: Accounting

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonSetGroupName xmlns="http://tempuri.org/">
  <TenantId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</TenantId>
  <GroupId>39484fd1-9103-4a0c-8924-ca2c3f0c1b89</GroupId>
  <GroupName>Accounting</GroupName>
</JsonSetGroupName>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": ""
}
```

6.c - Guest Account Functions

6.c.1 - Create a Guest User

Creates a guest account that can be used for sharing files/folders.

Request:

- Endpoint URL: /namespace/n.svc/jsoncreateguestuser
- HTTP-Method: POST
- Response Format: application/json
- Headers:
 - Content-Type: application/xml
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:
 - CreateGuestUser: XML Object containing the following properties.
 - DomainId: The domainid global unique identifier of the cluster administrator. Example: fee6f7ad-5cf6-4b55-9b4f-7400908be0ff
 - FirstName: The first name. Example: John
 - LastName: The last name. Example: Smith
 - Email: A valid email address. Example: user@example.com
 - Password: A password for the guest account. Example: MyP@ssw0rd!
 - SendWelcomEmail: A boolean to instruct the system to send a welcome email to the user. Example: true

```
<?xml version="1.0" encoding="UTF-8"?>
<CreateGuestUser xmlns="http://tempuri.org/">
<DomainId>fee6f7ad-5cf6-4b55-9b4f-7400908be0ff</DomainId>
<FirstName>John</FirstName>
<LastName>Smith</LastName>
<Email>user@example.com</Email>
<Password>MyP@ssw0rd!</Password>
<SendWelcomEmail>true</SendWelcomEmail>
</CreateGuestUser>
```

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "Guid": "9379e92d-167b-4552-978d-ac47f9595075",
  "LapseSeconds": 0
}
```

=====

7 - System Management

7.a - Diagnostic Functions

7.a.1 - Get Node Performance Information

Returns a snapshot of the current resources in use for a particular node. This information may be helpful to diagnose performance or outage problems.

Request:

- Endpoint URL: `/namespace/n.svc/jsonnodeperf`
- HTTP-Method: GET
- Response Format: `application/json`
- Headers:
 - `x-glad-token`: [Cookie value from jsonlogin here]
- Body:

Not applicable

Response:

```
{
  "Success": true,
  "Reason": "",
  "Context": "",
  "Context1": "",
  "Context2": "",
  "CacheInfo": "User List Count:2, Domain List Count:1, User Cache Count:3, allocated count:4",
  "RequestCount": 0,
  "TotalRequestCount": 299,
  "FireChangesQueueLength": 0,
  "HighPriorityDirCount": 2,
  "HighPriorityQueueDropCount": 0,
  "LowPriorityDirCount": 278,
  "LowPriorityQueueDropCount": 5,
  "PendingCompleteNodeRequests": 0,
  "PendingHighPriorityDirRequest": 0,
  "PendingLowPriorityDirRequest": 0,
  "PendingNodeRequests": 0,
  "PollStatusQueueLength": 0,
  "RegisteredClientCount": 0,
  "RequestTime": 1171,
  "SizeReportQueueLength": 0,
  "UploadQueueLength": 0,
  "WinUpgradeCount": 1,
  "NodeName": "node1"
}
```

7.a.2 - Get Cluster Performance Information

Returns a snapshot of the current resources in use for a particular node. This information may be helpful to diagnose performance or outage problems.

Request:

- Endpoint URL: /namespace/n.svc/clusterperf
- HTTP-Method: GET
- Response Format: application/xml
- Headers:
 - x-glad-token: [Cookie value from jsonlogin here]
- Body:

Not applicable

Response:

```
<GetClusterPerformanceResponse xmlns="http://tempuri.org/">
  <GetClusterPerformanceResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <Success xmlns="">false</Success>
    <Reason xmlns=""/>
    <Context xmlns=""/>
    <Context1 xmlns=""/>
    <Context2 xmlns=""/>
    <PerfMonCounters xmlns="">
      <NodePerfMon>
        <LastReportStr>6 seconds ago</LastReportStr>
        <CPU>0</CPU>
        <RequestTime>731</RequestTime>
        <Requests>0</Requests>
        <TotalRequests>310</TotalRequests>
        <NodeName>mypc</NodeName>
        <ChangeNotificationQLength>0</ChangeNotificationQLength>
        <PendingCompleteNodeRequestsLength>0</PendingCompleteNodeRequestsLength>
        <PendingNodeRequests>0</PendingNodeRequests>
        <PollStatusQueueLength>0</PollStatusQueueLength>
        <RegisteredClientCount>0</RegisteredClientCount>
        <PendingDirRequestInHighPriorityQueue>0</PendingDirRequestInHighPriorityQueue>
        <PendingDirRequestInLowPriorityQueue>0</PendingDirRequestInLowPriorityQueue>
      </NodePerfMon>
    </PerfMonCounters>
  </GetClusterPerformanceResult>
</GetClusterPerformanceResponse>
```

7.b - Miscellaneous Functions

7.b.1 - Check for Desktop Client Updates

Checks if a newer version of a desktop client has been published.

Request:

- Endpoint URL: /namespace/n.svc/jsoncheckupdate
- HTTP-Method: GET
- Response Format: application/json

- Headers:

- x-glad-token: [Cookie value from jsonlogin here]

- Body:

- JsonUpdateCheck: XML Object containing the following properties.
 - ProductType: One of the following predefined strings. Please make sure to include the colon at the end of the string: "WindowsClient:", "ServerAgent:", "MacClient:"
 - ProductVersion: The full version of your current client. Example: 9.1.2423.40381
 - Is64Bit: A boolean indicating whether the software is for a 64 bit system. If false, it is assumed to be for a 32 bit system. Example: true

```
<?xml version="1.0" encoding="UTF-8"?>
<JsonUpdateCheck xmlns="http://tempuri.org/"> <ProductType>WindowsClient:</ProductType>
<ProductVersion>8.10.2351.39808</ProductVersion>
<Is64Bit>true</Is64Bit>
</JsonUpdateCheck>
```

Response:

```
{
  "Success": true, "Reason": "", "Context": "", "Context1": "", "Context2": "",
  "PublishedVersion": "9.1.2421.40363",
  "TaskList":
  [
    {
      "Name": "replacewithzip",
      "Value": "http://node1.example.com/portal/Pkgs/windowspkg.zip",
      "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
      "LastAccessTime": "/Date(1517435244802-0500)/"
    },
    {
      "Name": "rename",
      "Value": "GladinetDDHandler64.dll\tGladinetDDHandler.dll", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
      "LastAccessTime": "/Date(1517435244802-0500)/"
    },
    {
      "Name": "rename",
      "Value": "GladinetShellProxy64.dll\tGladinetShellProxy.dll", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
      "LastAccessTime": "/Date(1517435244802-0500)/"
    },
    {
      "Name": "rename",
      "Value": "G1CopyHandler64.dll\tG1CopyHandler.dll", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
      "LastAccessTime": "/Date(1517435244802-0500)/"
    },
    {
      "Name": "rename",
      "Value": "G1OverlayIcon64.dll\tG1OverlayIcon.dll", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
      "LastAccessTime": "/Date(1517435244802-0500)/"
    },
  ],
  {

```

```
"Name": "rename",
"Value": "GLOverlayIconC64.dll\tGLOverlayIconC.dll", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
"LastAccessTime": "/Date(1517435244802-0500)/"
},
{
"Name": "rename",
"Value": "GLOverlayIconE64.dll\tGLOverlayIconE.dll", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
"LastAccessTime": "/Date(1517435244802-0500)/"
},
{
"Name": "rename",
"Value": "GLOverlayIconU64.dll\tGLOverlayIconU.dll", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
"LastAccessTime": "/Date(1517435244802-0500)/"
},
{
"Name": "rename",
"Value": "WosHiddenLauncher64.exe\tWosHiddenLauncher.exe", "NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000",
"Id": 0,
"LastAccessTime": "/Date(1517435244802-0500)/"
},
{
"Name": "exec",
"Value": "GladSetup.exe",
"NvGuid": "00000000-0000-0000-0000-000000000000", "UserGuid": "00000000-0000-0000-0000-000000000000", "Id": 0,
"LastAccessTime": "/Date(1517435244802-0500)/"
}
],
"UpdateStatus": "optionalupdate"
}
```