Authentication

Login to Keystone in OpenStack Swift.

URL structure
http://cloudspaces.urv.cat:5000/v2.0/tokens

Method
GET

Headers

- Content-Type: application/json

Request Body

```
{
    "auth":
    {
        "passwordCredentials":
        {
            "username": "userX",
            "password": "userX"
        },
        "tenantName": "userX"
    }
}
```

Response Body

```
{
    "access":{
        "token":{
            "issued_at":"2013-10-03T10:10:18.705131",
            "expires":"2013-10-04T10:10:18Z",
            "id":"MIIGEwYeMN-M=",
            "tenant":{
                "description":"Description userX",
                "enabled":true,
                "id":"0cd10527bafb4d778ba5a3a5bb419828",
                "name":"userX"
            }
        },
        "serviceCatalog":[
            {
                "endpoints":[
                    {
                        "adminURL":"https://cloudspaces.urv.cat:8080/
                        v1",
                        "region":"RegionOne",
```

```
                                        "internalURL":"https://cloudspaces.urv.cat:80
                                        80/v1/AUTH_0cd10527bafb4d778ba5a3a5bb419828",
                                        "id":"61807a44e9e141e998b67814bb1103c7",
                                        "publicURL":"https://cloudspaces.urv.cat:8080
                                        /v1/AUTH_0cd10527bafb4d778ba5a3a5bb419828"
                            }
                    ],
                    "endpoints_links":[
                    ],
                    "type":"object-store",
                    "name":"swift"
            },
            {
                    "endpoints":[
                            {
                                    "adminURL":"http://cloudspaces.urv.cat:35357/
                                    v2.0",
                                    "region":"RegionOne",
                                    "internalURL":"http://cloudspaces.urv.cat:500
                                    0/v2.0",
                                    "id":"1c77111ccbcc429a960d2bed06ea2942",
                                    "publicURL":"http://cloudspaces.urv.cat:5000/
                                    v2.0"
                            }
                    ],
                    "endpoints_links":[
                    ],
                    "type":"identity",
                    "name":"keystone"
            }
        ],
        "user":{
                "username":"userX",
                "roles_links":[
                ],
                "id":"85f0e8e483cb43e1b6a97d3cc27a58cb",
                "roles":[
                        {
                                "name":"_member_"
                        },
                        {
                                "name":"rusers"
                        }
                ],
                "name":"userX"
        },
        "metadata":{
                "is_admin":0,
                "roles":[
                        "9fe2ff9ee4384b1894a90878d3e92bab",
                        "a0eeaa6045c54d0d8e3dcf35c8aaf1ab"
                ]
        }
    }
}
```

Important values are:

● **access.token.id**: The access token that will be attached in every request to obtain access to API resources.

- **access.serviceCatalog**: It is a list that contains all services available. The one interesting for the API is Swift. Therefore, the only item on the list we need is the one that contains `swift` in the `name` parameter.
- **access.serviceCatalog[X].endpoints[0].publicURL**: The base URL where the API is located.

Error handling

Errors are returned using standard HTTP error code syntax. Any additional info is included in the body of the return call, JSON-formatted. Error codes not listed here are in the REST API methods listed below.

Standard API errors

Code	Description
400	Bad input parameter. Error message should indicate which one and why.
401	Authorization required. The presented credentials, if any, were not sufficient to access the folder resource. Returned if an application attempts to use an access token after it has expired.
403	Forbidden. The requester does not have permission to access the specified resource.
404	File or folder not found at the specified path.
405	Request method not expected (generally should be GET or POST).
5xx	Server error

Metadata

Retrieves file and folder metadata.

Method
GET

Headers

- StackSync-API: true
- X-auth-token: openstack_token

Parameters

- **file_id(Long) - (optional)** ID of a file or folder. If no ID is passed, it will return the metadata for the root folder.
- **list(Boolean) - (optional)** Only applicable when *fileId* corresponds to a folder. The strings *true* and *false* are valid values. *true* is the default. If *true*, the folder's metadata will include a contents field with a list of metadata entries for the contents of the folder. If *false*, the contents field will be omitted. Note that the root folder will always return its content regardless of this flag.
- **include_deleted(Boolean)**: (optional) The strings *true* and *false* are valid values, *false* is the default. If this parameter is set to *true*, then response will include metadata of deleted objects. Note that the target of the metadata call is always returned even when it has been deleted regardless of this flag.
- **version(Long)**: (optional) If you include a particular version, then only the metadata for that version will be returned.

Note that all parameters have to be URL-encoded.

Returns

The metadata for the file or folder at the given <file_id>. If <file_id> represents a folder and the list parameter is true, the metadata will also include a listing of metadata for the folder's contents.

Sample JSON return value for a file

```
{
    "is_folder":false,
    "mimetype":"application/pdf",
    "status":"CHANGED",
    "parent_file_id":0,
    "file_id":-534824681,
```

```
    "path":"/",
    "filename":"StackSync_Guide.pdf",
    "checksum":-2678858962222278590,
    "user":"Adrian",
    "client_modified":"2013-03-08 10:36:41.997",
    "size":775412,
    "version":2,
    "server_modified":"2013-03-08 10:36:41.997"
}
```

Sample JSON return value for a folder when list parameter is set to true.

If list is false the contents key will simply be omitted from the result.

```
{
    "is_root":false,
    "is_folder":true,
    "status":"NEW",
    "parent_file_id":0,
    "file_id":538757639,
    "path":"/",
    "filename":"Documents",
    "user":"Adrian",
    "client_modified":"2013-03-08 10:36:41.994",
    "version":1,
    "server_modified":"2013-03-08 10:36:41.994",
    "contents":[
        {
            "is_folder":false,
            "mimetype":"text/plain",
            "status":"NEW",
            "parent_file_id":538757639,
            "file_id":-1916666835,
            "path":"/Documents/",
            "filename":"important.txt",
            "checksum":-4653280656841610066,
            "user":"Adrian",
            "client_modified":"2013-03-08 10:36:42.893",
            "size":9453,
            "version":1,
            "server_modified":"2013-03-08 10:36:42.893"
        }
    ]
}
```

Return value definitions

field	description
is_folder	Whether the given entry is a folder or not
is_root	Whether the given entry is the root folder or not
status	Possible values are "NEW", "CHANGED", "DELETED". Indicating the status of the file in this specific version.
path	The canonical path to the file or directory
size	The file size in bytes (only for files)
mimetype	The media type of the file (only for files). http://www.iana.org/assignments/media-types
checksum	The file's checksum.
version	A unique identifier for the current version of a file. Can be used to detect changes and avoid conflicts.
filename	The name of the file or folder
user	The name of the user that made modified this specific version.
file_id	A unique identifier for a file or folder.
parent_file_id	file_id of the folder's parent
client_modified	This is the modification time set by the desktop client when the file was last modified, in the standard date format. Since this time is not verified (the server stores whatever the desktop client sends up), this should only be used for display purposes (such as sorting) and not, for example, to determine if a file has changed or not.
server_modified	This is the modification time set by the server at the time of processing the file.

Note that *status, parent_file_id, client_modified, server_modified, version, user, path, checksum* are not in the metadata for the root folder.

Status codes

Status code	Description
200-299	The request was successful. The folder information is in the response body.
500-599	Server error

See also standard error codes.

Versions

Retrieve file versions.

Method
GET

Headers

- StackSync-API: true
- X-auth-token: openstack_token

Parameters

- **file_id(Long) - (obligatory)** ID of a file or folder.
- **limit_versions(long): no implemented yet.**

Note that all parameters have to be URL-encoded.

Returns

The same information as in the Metadata call

Sample JSON return value for a file

```json
{
    "file_id":-7177933568794312281,
    "parent_file_id":0,
    "filename":"config.xml",
    "path":"",
    "is_folder":false,
    "status":"CHANGED",
    "server_modified":"2013-06-18 10:44:56.958",
    "client_modified":"2013-06-18 10:44:53.571",
    "user":"gguerrero201306180926",
    "version":3,
    "checksum":29300690,
    "size":2798,
    "mimetype":"application/xml",
    "versions":[
        {
            "file_id":-7177933568794312281,
            "parent_file_id":0,
            "filename":"config.xml",
            "path":"",
            "is_folder":false,
```

```
        "status":"CHANGED",
        "server_modified":"2013-06-18 10:44:56.958",
        "client_modified":"2013-06-18 10:44:53.571",
        "user":"gguerrero201306180926",
        "version":3,
        "checksum":29300690,
        "size":2798,
        "mimetype":"application/xml"
    },{
        "file_id":-7177933568794312281,
        "parent_file_id":0,
        "filename":"config.xml",
        "path":"",
        "is_folder":false,
        "status":"CHANGED",
        "server_modified":"2013-06-18 10:44:51.258",
        "client_modified":"2013-06-18 10:44:48.251",
        "user":"gguerrero201306180926",
        "version":2,
        "checksum":978130575,
        "size":2790,
        "mimetype":"application/xml"
    },{
        "file_id":-7177933568794312281,
        "parent_file_id":0,
        "filename":"config.xml",
        "path":"",
        "is_folder":false,
        "status":"NEW",
        "server_modified":"2013-06-18 09:49:04.946",
        "client_modified":"2013-06-18 09:28:55.24",
        "user":"gguerrero201306180926",
        "version":1,
        "checksum":3189249362,
        "size":2782,
        "mimetype":"application/xml"
    }]
}
```

Restore file

Restore a file.

https://domain.ext/stacksync/restore

POST

- StackSync-API: true
- X-auth-token: openstack_token

- **file_id:** *(required)* The file id.
- **version:** (*required*) The file version**.**

Note that all parameters have to be URL-encoded.

Metadata of the file restored.

Sample JSON return value

```
{
    "is_folder":false,
    "mimetype":"application/pdf",
    "status":"CHANGED",
    "parent_file_id":0,
    "file_id":-534824681,
    "path":"/",
    "filename":"StackSync_Guide.pdf",
    "checksum":-2678858962222278590,
    "user":"Adrian",
    "client_modified":"2013-03-08 10:36:41.997",
    "size":775412,
    "version":2,
    "server_modified":"2013-03-08 10:36:41.997"
}
```

Upload a file

Uploads a file.

Method
PUT

Headers
- StackSync-API: true
- X-auth-token: openstack_token

Parameters
- **file_name:** *(required)* The file name.
- **parent:** (optional) ID of the folder where the file or folder is going to be uploaded. If no ID is passed, it will use the top-level folder. This parameter should **not** point to a file.
- **overwrite: (not available yet)** This value, either true (default) or false, determines what happens when there's already a file with the same name at the specified path. If true, the existing file will be overwritten by the new one. If false, the new file will be automatically renamed (for example, test.txt might be automatically renamed to test (1).txt). The new name can be obtained from the returned metadata.

Note that all parameters have to be URL-encoded.

Returns
Metadata of the file just uploaded.

Sample JSON return value

```
{
    "is_folder":false,
    "mimetype":"application/pdf",
    "status":"CHANGED",
    "parent_file_id":0,
    "file_id":-534824681,
    "path":"/",
    "filename":"StackSync_Guide.pdf",
    "checksum":-2678858962222278590,
    "user":"Adrian",
    "client_modified":"2013-03-08 10:36:41.997",
    "size":775412,
    "version":2,
    "server_modified":"2013-03-08 10:36:41.997"
```

```
}
```

Status codes

Status code	Description
200-299	The file has been successfully uploaded
500-599	Server error

See also standard error codes.

Download a file

Retrieves a file's content.

URL structure

https://domain.ext/stacksync/files

Method

GET

Headers

- StackSync-API: true
- X-auth-token: openstack_token

Parameters

- **file_id:** *(required)* ID of the file to download.
- **version**: (optional) If you want to download a specific version.

Note that all parameters have to be URL-encoded.

Returns

The content of the file in the body response.

Delete a file or folder

Method

DELETE

Headers

- StackSync-API: true
- X-auth-token: openstack_token

Parameters

- **file_id:** *(required)* ID of the file or folder to delete.

Note that all parameters have to be URL-encoded.

Returns

Metadata of the file just uploaded.

Create a folder

Touch a folder in the server.

Method
POST

Headers

- StackSync-API: true
- X-auth-token: openstack_token

Parameters

- **folder_name:** *(required)* Name of the folder to be created.
- **parent:** (optional) ID of the folder where the folder is going to be created. If no ID is passed, it will use the top-level folder. This parameter should **not** point to a file.

Note that all parameters have to be URL-encoded.

Returns

Metadata of the file just uploaded.

Sample JSON return value

```
{
    "is_folder":true,
    "mimetype":"inode/directory",
    "status":"CHANGED",
    "parent_file_id":0,
    "file_id":-534824681,
    "path":"/",
    "status": "NEW",
    "filename":"folder1",
    "checksum":-2678858962222278590,
    "user":"Adrian",
    "client_modified":"2013-03-08 10:36:41.997",
    "size":0,
    "version":2,
    "server_modified":"2013-03-08 10:36:41.997"
}
```

Status codes

Status code	Description
200-299	The folder has been successfully created
400	Folder already exists
500-599	Server error

See also standard error codes.