



Syncplicity®
Content API
Document Version 1.3.11

Syncplicity believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS". NEITHER SYNCPLICITY, LLC, NOR ITS AFFILIATES, MAKES ANY REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, or distribution of any Syncplicity software described in this publication requires an applicable software license. For the most up-to-date listing of Syncplicity product names, see www.syncplicity.com. All trademarks used herein are the property of their respective owners.

Syncplicity Content API Reference Guide

Version	Publication	Notes
1.3.11	April 2017	<ul style="list-style-type: none"> • Added New parameter in Syncpoint Get • Removed "As-User" attribute • Fixed Typo
1.3.10	February 2017	<ul style="list-style-type: none"> • Added new Folder Type in Syncpoint Post, Put and Get definitions list. • Changed URL for PII service
1.3.9	December 2016	<ul style="list-style-type: none"> • Added information and changes about ROL/GOL awareness
1.3.8	November 2016	<ul style="list-style-type: none"> • Updated API documentation for values that are no longer Integer, but should be represented as Long due to Integer-overflow concerns. <ul style="list-style-type: none"> ○ Updated APIs: <ul style="list-style-type: none"> ▪ Syncpoints POST response RootFolderId is Long instead of Integer ▪ Syncpoints GET response RootFolderId is Long instead of Integer ▪ Syncpoint GET response RootFolderId is Long instead of Integer ▪ Syncpoint PUT input RootFolderId is Long instead of Integer ▪ Syncpoint PUT response RootFolderId is Long instead of Integer ▪ Folders POST url-param FOLDER_ID is Long instead of Integer ▪ Folders POST response FolderId is Long instead of Integer ▪ Folders GET url-param FOLDER_ID is Long instead of Integer ▪ Folders GET url-param LAST_FOLDER_ID is Long instead of Integer ▪ Folders GET response FolderId is Long instead of Integer ▪ Folders GET response Folders.FolderId is Long instead of Integer ▪ Folders DELETE url-param FOLDER_ID is a Long instead of Integer

Version	Publication	Notes
1.3.7	October 2016	Removed reference to reports
1.3.6	September 2016	Added API limits
1.3.5	September 2016	<ul style="list-style-type: none"> Added descriptive language for Rate Limiting
1.3.4	August 2016	<ul style="list-style-type: none"> Updated document footer from "Migration API" to "Content API". Updated architecture section illustrating differences between on prem and cloud storage (available in the US and EU sites). Minor typographical updates. Added descriptive intro to the Migration Programming flow. Added version specific steps for on-prem Storage Vaults prior to v2.7.0.5 Re-Titled this document to reference the "version" of the document, not the "version" of the various Syncplicity components that are separately versioned such as the client and storage connector. Updated references to folder re-sharing rights capabilities
1.3.3	June 2016	<ul style="list-style-type: none"> Added Versions API
1.3.2	June 2016	<ul style="list-style-type: none"> Revisions
1.3.1	May 2016	<ul style="list-style-type: none"> Update with Files Get
1.3	May 2016	<ul style="list-style-type: none"> Update to Folders Get Added Links APIs
1.2	Mar. 2016	<ul style="list-style-type: none"> Overview of Content Migration section updated. Added As-User to HTTP Header Parameters.
1.1	Mar. 2016	<ul style="list-style-type: none"> Added missing JSON response to Syncpoint Participants DELETE.
1.0	December 2015	<p>Production release. Changes include:</p> <ul style="list-style-type: none"> Programming flow (end of Chapter 1): Added a final step to add users/groups to shared folders. Various commands: Removed 0: unknown type, and 7: private, as valid values for Type of syncpoint; updated the description. Various commands: Removed "Note: '11111111-1111-1111-1111-111111111111' is also valid value" from the Valid Values of StorageEndpointID. Various commands, JSON examples: Replaced number for IDs with a variable, <param ID>. Storage Endpoint Get and Syncpoints Get: Changed {USER_GUID} param name to user={USER_GUID} Storage Endpoint Get: Removed auths and AccessKey; added ProtocolVersion. Syncpoints GET: Removed Participants and SharingInviteNote. Syncpoint GET: Corrected JSON example to single object notation instead of array notation. Removed Participants, RemoteWipe, ServerID and SharingInviteNote. Syncpoint PUT: Replaced JSON examples, which were in array notation instead of single object notation. Removed

Version	Publication	Notes
		Parent, Permission, RemoteWipe, Participants, ServerID and SharingInviteNote. <ul style="list-style-type: none"> • Folders POST: Added a note to Folder_ID URL parameter. • Upload File (saveFile POST): Updated the file chunks and multi-part body descriptions; added example for FILE_PATH. • Download File (retrieveFile GET): Added example for V_TOKEN, and a note for Range. Added a response definition. • Syncpoint Participants POST: Updated resource URL, added URL query parameters.
0.9.1	Oct. 2015	Publication for review only.

Chapter 1 Overview	11
Prerequisites for developing with Syncplicity APIs	12
First-time login to Syncplicity Developer Portal	13
Working with your sandbox account.....	15
What you can do	15
What you cannot do	15
Understanding Throttling	15
Global Syncplicity	16
What is the Regional Orchestration Layer - ROL.....	16
What is the Global Orchestration Layer - GOL	17
Does my application need to be ROL/GOL aware?.....	17
What changes are needed to be ROL/GOL aware.....	17
What if you don't need to be ROL/GOL aware	18
Understanding Syncplicity Storage	19
Syncplicity or Third Party Hosted Cloud Storage Configuration	20
On-Premise Storage Connector configuration.....	20
Overview of Content Migration.....	22
Setup Procedure	22
Programming Flow	23
Chapter 2 Content APIs	25
Storage Endpoint GET	25
Resource URL	25
URL Parameters.....	25
HTTP Header Parameters.....	26
Sample JSON Response	27
Sample JSON Response Definitions	27
Response Error Details.....	29
Syncpoints POST	31
Resource URL	31
HTTP Header Parameters.....	31
Sample JSON Input	32
Sample JSON Input Definitions	32

Sample JSON Response	33
Sample JSON Response Definitions	34
Response Error Details.....	38
Syncpoints GET (Optional)	39
Resource URL	39
URL Parameters.....	39
HTTP Header Parameters.....	40
Sample JSON Response	41
Sample JSON Response Definitions	41
Response Error Details.....	46
Syncpoint GET.....	47
Resource URL	47
URL Parameters.....	47
HTTP Header Parameters.....	47
Sample JSON Response	48
Sample JSON Response Definitions	49
Response Error Details.....	51
Syncpoint PUT	53
Resource URL	53
HTTP Header Parameters.....	53
Sample JSON Input	54
Sample JSON Input Definitions	55
Sample JSON Response	57
Sample JSON Response Definitions	57
Response Error Details.....	59
Syncpoint DELETE (Optional)	61
Resource URL	61
URL Parameters.....	61
HTTP Header Parameters.....	61
Response Error Details.....	62
Folders POST	63
Resource URL	63
URL Parameters.....	63
HTTP Header Parameters.....	64
Sample JSON Input	64
Sample JSON Input Definitions	65
Sample JSON Response	65
Sample JSON Response Definitions	65
Response Error Details.....	66

Folders GET	67
Resource URL	67
URL Parameters.....	67
URL Query Parameters.....	68
HTTP Header Parameters.....	68
Sample JSON Response	69
Sample JSON Response Definitions	69
Response Error Details.....	71
Folder DELETE (Optional)	72
Resource URL	72
URL Parameters.....	72
URL Query Parameters.....	72
HTTP Header Parameters.....	73
Upload File [saveFile POST]	74
Resource URL	75
HTTP Headers Parameters	75
URL Query Parameters.....	76
Form Data	76
Sample Form Data	77
Sample Response.....	77
Response Definitions	77
Response Error Details.....	78
Download File [retrieveFile GET] (Optional).....	79
Resource URL	79
URL Query Parameters.....	79
HTTP Header Parameters.....	79
Response	80
Response Error Details.....	80
Files GET	81
Resource URL	81
URL Parameters.....	81
HTTP Header Parameters.....	82
Sample JSON Response	83
Sample JSON Response Definitions	83
File DELETE (Optional)	85
Resource URL	85
URL Parameters.....	85
URL Query Parameters.....	85
HTTP Header Parameters.....	85

Response Error Details.....	86
Syncpoint Participants GET.....	87
Resource URL	87
URL Parameters.....	87
HTTP Header Parameters.....	87
Sample JSON Response	88
Sample JSON Response Definitions	89
Response Error Details.....	89
Syncpoint Participants POST.....	90
Resource URL	90
URL Parameters.....	90
URL Query Parameters.....	90
HTTP Header Parameters.....	91
Sample JSON Input	91
Sample JSON Input Definitions	92
Sample JSON Response	93
Sample JSON Response Definitions	93
Response Error Details.....	94
Syncpoint Participants DELETE	95
Resource URL	95
URL Parameters.....	95
HTTP Header Parameters.....	95
Sample JSON Input	96
Sample JSON Input Definitions	96
Links POST	97
Resource URL	97
HTTP Header Parameters.....	97
Sample Input Definition	97
Sample JSON Input	98
Sample JSON Response definitions	99
Sample JSON Output.....	101
Response Error Details.....	101
Links GET	102
Resource URL	102
URL Parameters.....	102
HTTP Header Parameters.....	102
Sample JSON Response definitions	103
Sample JSON Output.....	105

Link PUT	106
Resource URL	106
URL Parameters.....	106
HTTP Header Parameters.....	106
Sample Input Definition	107
Sample JSON Input	107
Sample JSON Response definitions	107
Sample JSON Output.....	109
Response Error Details.....	109
LinkGET	110
Resource URL	110
URL Parameters.....	110
HTTP Header Parameters.....	110
Sample JSON Response definitions	111
Sample JSON Output.....	112
Response Error Details.....	113
Link DELETE.....	114
Resource URL	114
URL Parameters.....	114
HTTP Header Parameters.....	114
Response Error Details.....	115
Versions GET	116
Resource URL	116
URL Parameters.....	116
HTTP Header Parameters.....	116
Sample JSON Response	117
Sample JSON Response Definitions	117
Response Error Details.....	118
Version DELETE	119
Resource URL	119
URL Parameters.....	119
HTTP Header Parameters.....	119
Response Error Details.....	120
Chapter 3 API limits	121
Chapter 4 ROL / GOL Awareness	123
OAuth Authentication.....	123
ROls Get.....	124
Resource URL	124

URL Parameters.....	124
HTTP Header Parameters.....	125
Personal Identifiable Information (PII)	125
Sample JSON Response	127
Users PII GET.....	128
Resource URL	128
HTTP Header Parameters.....	128
Sample JSON Input	129
Sample JSON Input Definition.....	129
Sample JSON Response	130
Sample JSON Response Definitions	130
Response Error Details.....	131

Chapter 1 Overview

This Syncplicity Content API documentation provides details on each of the API calls required to create, read, update or delete a user's content to Syncplicity. Here is the list of things you need to do before you start developing your content application using Syncplicity APIs:

1. Make sure you have a Syncplicity Enterprise Edition account you can use to login to the Developer Portal at <https://developer.syncplicity.com>.
2. Setup your developer sandbox account by logging into the Developer Portal which will prompt you to configure your sandbox account password.
3. Login to your developer sandbox account by visiting <https://my.syncplicity.com> to make sure its correctly provisioned and that you can access it.
4. On the account tab of your developer sandbox account, create an "Application Token" that you will use to authenticate before making API calls.
5. Download sample code to get a feel for developing code against our APIs. We have sample code available in C# and Java.
6. Register your app in the Developer Portal to obtain the "App Key" and "App Secret".
7. Now start writing some code...

Prerequisites for developing with Syncplicity APIs

In order to start using the Syncplicity API, you need to meet the following prerequisites:

- **Understand Syncplicity Features:** Before you start using the Syncplicity APIs, it is important that you understand the overall administrative and end-user functionalities of the Syncplicity File Sync and Share service. To learn more about Syncplicity features, please check out our [website](#) and our [online documentation](#).
- **Syncplicity Enterprise Edition account:** In order to log into Syncplicity Developer portal, a developer needs to be a member of a Syncplicity Enterprise Edition account.
- **Syncplicity Administrator user account:** In order to work with the Syncplicity API you will need to create a Global Administrator user (directly in the Syncplicity Admin Console) in the company account where content migration will take place. This user will be used as the Admin user, and you should login as this user and create an Application Token on the account tab. If your organization synchronizes user identities from Active Directory (AD) or LDAP to Syncplicity, this Admin user should not be created in the AD or LDAP directory.
- **Syncplicity Application Token:** In order to authenticate and authorize your app using the "client credentials" OAuth grant type for use with the Syncplicity Orchestration APIs, you need to have the Syncplicity Application Token for your Migration Admin user handy.
- **Syncplicity Migration Administrator user device:** In order to work with the On-Premise Storage Connector API you will need to create a user device under the Admin user's account that will be used for migration purposes. During the device creation process you will receive the device's machine token.
- **Syncplicity Machine Token:** In order to authenticate and authorize your app using the "client credentials" OAuth grant type, for use with the Syncplicity On-Premise Storage Connector APIs, you will need to have the Syncplicity Machine Token for your Admin user handy.
- **Experience in developing using REST APIs:** The Syncplicity APIs follow the REST architectural style and supports JSON for passing data in requests and for receiving data in responses.

First-time login to Syncplicity Developer Portal

You can log into Syncplicity Developer Portal using your Syncplicity login credentials. **Only Syncplicity Enterprise Edition users are allowed to login to the Developer Portal.** Based on the configuration done by your Syncplicity administrator, the Syncplicity Developer Portal will present one of the following options for login:

- Basic Authentication using Syncplicity username and password.
- Enterprise Single Sign-on using the Web-SSO service used by your organization.

Once you have successfully logged in for the first time, the Syncplicity Developer Portal automatically creates an Enterprise Edition sandbox account to help you develop and test your application. Here is how it works:

Set Password for your API Development Sandbox Account

Welcome to Syncplicity API Developer Portal!

To simplify and accelerate your API development process, we have created an Enterprise Edition sandbox account for you with the email address - (ak.dp-apidev@dispostable.com). Please note that this is a special email address that was generated by "-apidev" to your Syncplicity account email address.

- You cannot log into the developer portal using this email.
- You can log into this Syncplicity EE sandbox account online by going to my.syncplicity.com site and using this sandbox email address along with the password you are about set below.

To complete sandbox account creation, please set your password for this sandbox account.

Password

Confirm password Passwords match: yes

Password must contain a minimum of 8 characters including an uppercase letter, a lowercase letter, and a number/symbol.

- The Syncplicity Developer Portal automatically creates your sandbox account by appending "**-apidev**" to the email address you used for logging into the Developer Portal. For e.g. if you logged into Syncplicity Developer Portal using **user@domain.com** as your email address, then your associated sandbox account username is **user-apidev@domain.com**.
- The Developer Portal will prompt you to set your password for this sandbox account. See screenshot above for reference.
- After you have successfully setup your password, you can use the sandbox email address and the newly configured password for logging into your sandbox account by visiting <https://my.syncplicity.com> and using the username that includes "**-apidev**". So, in the example above, you will have to use "user-apidev@domain.com" as the username to log in to your sandbox account.

Working with your sandbox account

Once you have successfully created your sandbox account (as described in the previous section), you can begin testing your app by making API calls against the sandbox account. You are the owner of this sandbox Enterprise Edition account and have administrative privileges over it.

What you can do

The developer sandbox account provides the following capabilities:

- The sandbox account is a FREE Enterprise Edition account for developing applications, and has the following configuration:
 - 3-years lease to enable development. To extend it, you can call customer support.
 - 10 GB of content storage capacity.
 - Up to 1000 user seats.
- You can take all actions that that a Syncplicity administrator can perform on the account. For information on administrative controls, check out our [online documentation](#).

What you cannot do

Like any other Enterprise Edition account, you cannot do the following:

- Increase the max limit for user seats and content storage capacity.
- Increase the lease period of the sandbox account.

Understanding Throttling

- Currently the API gateway exposes throttling behavior of 15/calls per second and 10,800 calls per hour. From a coding perspective, the API gateway will return a 429 Too Many Requests error if the limit is reached. The expectation is that if you wait a moment to retry, then the clock used to calculate the quota/spike arrest over a given period of time will have progressed taking you below the limit so that you can successfully call the API on the subsequent attempt. This window of time to delay API calls is a preferred best practice.

Global Syncplicity

As Syncplicity continues to move forward with a global service solution and as security and privacy concerns continue to rise, Syncplicity will now support multiple data centers for storing and managing the company and user metadata that can reside in isolation from other data centers. This allows Syncplicity to fulfill the privacy and security concerns regarding Safe Harbor and similar concerns by storing and processing the end-user's company data and requests within the geographical region. Initially, Syncplicity will support the US-only region via the existing data center and a new data center will be implemented in the European Union (EU) to fulfill the Safe Harbor requirements for EU.

What is the Regional Orchestration Layer - ROL

During the initial company/account creation, it will be necessary for the company to choose which data center (ROL) will become the primary data center. Currently, the US and EU are the two regions supported in this manner. Once the company is created within the Syncplicity system, however; there is no ability to change which region is the home region. It is this ROL that will provide access to the PII related metadata for each employee of that company, therefore APIs should be directed to the home ROL for looking up the PII for each user of that company, the other ROL(s) will return errors if the API is used in the incorrect ROL.

If you as a developer need to know the hostname of the API gateway associated with the two current ROLs, they are cataloged below:

API Gateway Host	ROL	ROL ID	Comments
api.syncplicity.com	US	1	This is the ROL to use for US regional companies that have explicitly chosen the US region as their data center during Syncplicity account/company creation. It is also the ROL to use for all existing applications that are not ROL aware.
api.eu.syncplicity.com	EU	2	This is the ROL to use for EU regional companies that have chosen EU as the data center explicitly during Syncplicity account/company creation.

The ROL Id is important to note as several services that return User information such as Syncpoint Participants API, will return a list of users and the ROL Id that they reside

in. This is needed to look up personal information (First Name, Last Name, Email) of that user if the ROL that the specified user resides is not the home ROL for the application.

What is the Global Orchestration Layer - GOL

The global orchestration layer is designed to help the API consumer to get information about ROLs and understand which ROL is used as the storage for a given user. There is one primary read-only GET REST requests, which should allow API consumer to understand where a given user's home ROL is located. The GOL is a public service which doesn't require any authorization.

In addition to informing the developer which ROL(s) a given user participates (has Syncpoints, shared links etc.) it also informs the developer which ROL is the home ROL (where the PII data is stored) as well as the hostname of the API gateway for that given ROL. It is important to note that all API calls have the same endpoint and parameters between each of the ROLs, but the hostname of the API gateway that handles that specific ROL is the only notable change to the URL/API call construction when compared to non-ROL aware API calls historically.

Does my application need to be ROL/GOL aware?

It is important to note that existing applications which is written against the Syncplicity API gateway does not need to change. The only reason to update or write a new application that is GOL and ROL aware, if the developer knows that there is PII that needs to be retrieved which resides in a different data center ROL.

For example, if the developer is aware that there are both home ROL and remote ROL participants in a single Syncpoint (i.e. both the US and EU ROL users share the same Syncpoint) and the user's name and email of the remote user are needed, then the developer should read the section "[What changes are needed to be ROL/GOL aware](#)" below and understand about ROL/GOL integration in more depth.

It is important to note that the above use case is just an example and is not the only use case where users may involve users from multiple ROLs and if a company is head quartered in a particular region it does not mean that they reside in that region's ROL. There are many EU customers that are based out of the US ROL. So developer should be aware of when they need to be ROL aware in their coding.

What changes are needed to be ROL/GOL aware

There are a number of changes that the application developer will need to be aware to integrate fully with the multiple ROL concept and the GOL and PII service look up service calls. These are described in full in the subsequent chapter [ROL / GOL Awareness](#).

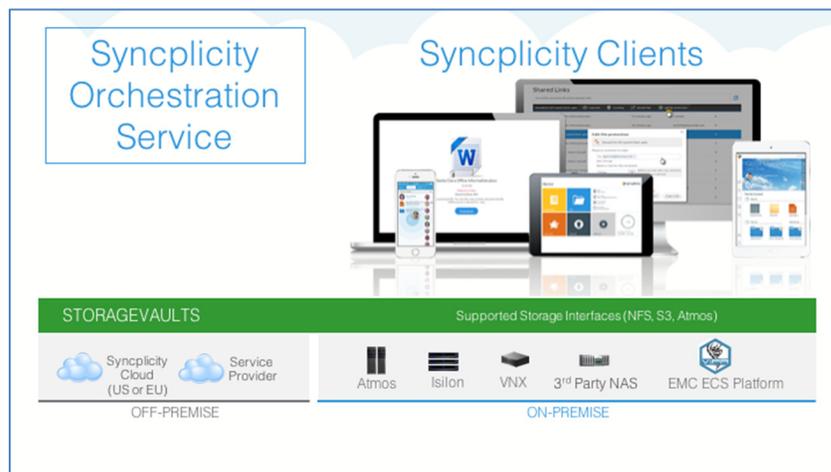
What if you don't need to be ROL/GOL aware

As a developer, if your application does not need to be ROL/GOL aware, the only thing you will need to know is which hostname to invoke for the APIs of your home ROL. If the application was written prior to the ROL/GOL capability, then no changes are needed, continue invoking the APIs using `api.syncplicity.com`. If you are new developer writing a new application, then you will need to know which ROL is the company's home ROL. Based on that, the host name is one of the ROL host names listed below in the table:

API Gateway Host Name	ROL	ROL ID
<code>api.syncplicity.com</code>	US	1
<code>api.eu.syncplicity.com</code>	EU	2

Understanding Syncplicity Storage

The Syncplicity logical architecture is comprised of three primary components referenced in the following diagram:



- A cloud-based **orchestration layer** (top right in the above diagram) that controls the sync process, enabling sharing of files and folders between users and devices. This is a multi-tenant cloud-based service that is common across all Syncplicity customers.
- **Syncplicity clients** (top left in the diagram above) packaged software available from Syncplicity that is installed on Windows or Mac Desktops, or available from any supported web browser, or installed on any supported mobile platform. The clients represent the presentation layer visible to the end user where files and folders can be added to Syncplicity. Once a folder is added to Syncplicity it is the actual physical storage that the StorageVaults point to and where files are actually stored.
- **Syncplicity StorageVaults** (bottom left and right in the diagram above) control where files (and past file versions) are stored in Syncplicity. This layer is single-tenant for customers that choose on premise storage and is multi-tenant for customers using Syncplicity's public cloud storage.

Syncplicity or Third Party Hosted Cloud Storage Configuration

Syncplicity offers hosted multi-tenant cloud storage (off premises) in both the US and EU regions. Alternatively, Syncplicity has partnered with Service Providers who will host a multi-tenant shared or single-tenant dedicated StorageVaults. Contact your Syncplicity Customer Success representative for more details about each.

On-Premise Storage Connector configuration

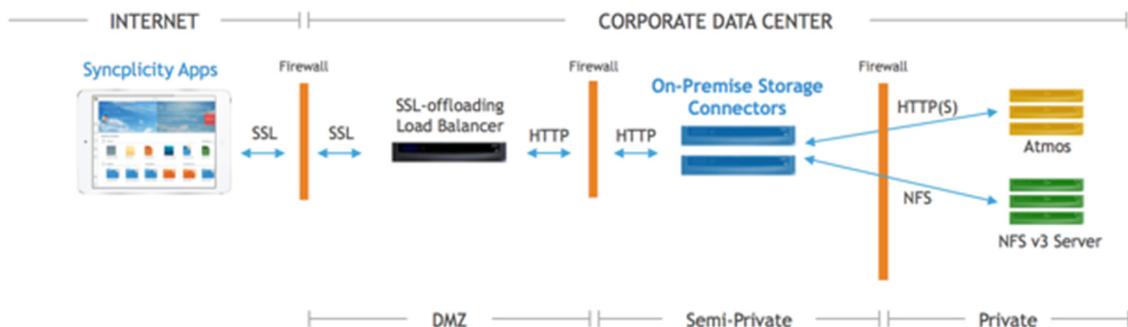
For customers choosing to deploy a private, single tenant, instance of storage on premises, you can deploy the Storage Connector in your own data center. In this configuration, the customer is responsible for providing the infrastructure necessary to store end user files in a StorageVault.

The network configuration requirements for on-premise storage include an externally-addressable SSL-offloading load balancer in front of all virtual machines, a storage backend that supports a standard NFS v3 interface or an EMC Atmos (v2.1+) storage system, and various inbound and outbound open ports.

To configure on-premise storage, you need to install the Syncplicity On-Premise Storage Connector. The Storage Connector is initially supplied as an OVA file and installed on a virtual machine, and upgrades are provided as RPM files. Each Storage Connector requires its own virtual machine hosted on VMware vSphere Hypervisor (i.e., ESXi).

Additionally, it is required that you deploy an externally-addressable SSL-offloading load balancer in front of all virtual machines, configured with a Certificate Authority (CA) signed (NOT self-signed) SSL certificate.

As shown in the diagram, a typical example is that the storage layer is in the private area of the corporate network, the Storage Connectors virtual machines are in the semi-private area, and the SSL-offloading load balancer is in the DMZ.



The Storage Connector requires specific inbound and outbound ports to be open, as specified in the On-Premise Storage Connector article, [Prerequisites](#). If using StorageVault Authentication, refer to the *Syncplicity StorageVault with Authentication Configuration Guide* for configuration procedures.

You should deploy the most recent version of the Syncplicity Storage Connector software as referenced in the prerequisites section. All Storage Connector versions newer than v2.5.2.0 (Released in December 2015) are compatible with the Content API. Storage Connector v2.7.0.5 (Released in July 2016) is automatically configured to communicate with the Syncplicity API gateway. If your corporate Storage Administrator has not yet upgraded to v2.7.0.5 you will require the following manual configuration changes to all your storage connectors to connect to the Syncplicity API gateway.

Verify that the external URL to the Syncplicity API gateway is enabled on each Storage Connector, as follows:

1. At the virtual machine, edit the following file using the vi editor:

```
sudo vi /etc/syncp-storage/syncp-storage.conf
```

2. Enter these two "syncplicity.ws.external" lines as new rows and save the file:

```
syncplicity.ws.external.enabled: true  
syncplicity.ws.external.url: "https://api.syncplicity.com"
```

3. Enter this command to restart the syncplicity service:

```
service syncp-storage restart
```

Overview of Content Migration

A common use case for exposing a content based API is to bulk migrate content from existing data sources such as file servers or end point desktops into Syncplicity. The following section provides an illustration of how a custom-built migration application would be designed, and how its data would flow into Syncplicity.

Setup Procedure

1. Deploy a StorageVault either hosted in a Syncplicity cloud or using an On-Premise Storage Connector and configure external URL settings, as described in the previous section.
2. Provision a Global Administrator as the migration administrator and change their "Access only own content through API" privilege to "Access content on behalf of managed users through API" on the Manage User page. Then obtain this administrator's application token from their account page.
3. Use the curl command to create a migration "device" in the migration administrator's account and save a copy of the migration device's ID and machine token.

```
curl -X POST https://xml.syncplicity.com/1.1/auth/machines.svc/ -H 'Authorization: Application <migration_admin_application_token>' -H 'Accept: application/json' -H 'Content-Type: application/json' -d '{"Name": "API-migration", "SystemName": "Windows", "SystemVersion": "9.9.9", "ClientVersion": "9.9.9"}'
```

4. Use a Firefox browser to retrieve the StorageVault Authentication storage token.
 - a. Turn on the Firefox developer network analysis tool by pressing the F12 key.
 - b. Use Firefox to navigate to this URL after replacing the example values with real ones:

```
https://<storage_endpoint_url>/v2/auth?machine_id=<machine_device_id>&scheme=syncpssso253751160://
```

- c. You will be re-directed to your company's identity provider single sign-on (SSO) page for authentication.
- d. After authentication, the browser will be re-directed to

```
https://<storage_endpoint_url>/v2/auth/acs
```

- e. Nothing will display on the browser screen – use the Firefox developer network tool to extract the value of the 'Location' HTTP response header.
- f. Use a URL decoder to decode the storage token (the part after 'syncpssso253751160://token/'). Note that the storage token starts with 'SSOUser' and this is part of the token.

5. Use the Syncplicity Developer Portal to create an app.

Programming Flow

For details, see the Developer Portal Getting Started Guide (online) and provided sample code.

1. Use the app key, app secret and migration administrator's application token to call the Syncplicity OAuth service to obtain a user bearer token.
2. Use the app key, app secret and migration administrator's machine token to call the Syncplicity OAuth service to obtain a machine bearer token.
3. For each target migration user, call the User Service to obtain the user's GUID.
4. For each target migration user, call the Storage Endpoint service to obtain the user's default StorageVault.
5. There are two methods for migrating a user's content:
 - a. Migrate the content into the migration administrator's account but store it in the target migration user's default StorageVault. Then when the migration is complete, transfer ownership of the syncpoints from the migration administrator to the target migration user. This approach has the advantage that the user will not receive the migrated content until the migration is complete.
 - b. Use the "As-User" HTTP header (the Administrator making the API calls must have been assigned the corresponding privilege via the Syncplicity admin console) and perform the migration directly on-behalf-of the target migration user. This approach should be used for incremental migrations. If used for an initial migration, it has the disadvantage of the user receiving the migrated content during the migration process instead of when the process is complete.
6. For each target migration user, use either method 5a or 5b, and iterate through:
 - a. Create a syncpoint on the user's default StorageVault.
 - b. Recursively create subfolders and upload files in each folder, including the root folder.
7. If using method 5a, when all of a target migration user's content has been migrated:
 - a. Change the ownership of each of the user's syncpoints to the target migration user.
 - b. Remove the migration administrator as a participant (sharee) on each of the user's syncpoints.

- c.** For shared folders (syncpoints), add the appropriate groups and/or individual users as sharees.
- 8.** If performing an incremental migration of content that has been updated on the source system, use method 5b.

Chapter 2 Content APIs

This chapter lists all the Syncplicity Content APIs.

Storage Endpoint GET

Retrieves metadata and details about the default storage endpoint (StorageVault) for a given user as identified by the user's GUID.

Resource URL

https://api.syncplicity.com/storage/storageendpoint.svc/?user={USER_GUID}

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
user={USER_GUID}	No	If specified, retrieves the storage endpoint (StorageVault) definitions of the specified user; otherwise, the current user of the application is used.	GUID	Application must be working on behalf of an administrator user when specifying USER GUID for a different user.

HTTP Header Parameters

Name	Value	Description
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (optional)	application/json	Content-Type of the request

Sample JSON Response

```
{
  "Id": "<GUID Identifier of the StorageVault>",
  "Name": "<Name of the StorageVault>",
  "Description": "<Description>",
  "CompanyId": "<GUID Identifier of the Company>",
  "Active": <status true or false>,
  "Urls": [
    {
      "Url": <Url of the StorageVault e.g. "https://10.10.10.100:9000">
    }
  ],
  "Auths": [
    {
      "AccessKey": "<datastore access key>"
    }
  ],
  "Version": 0,
  "SizeGb": <Storage Capacity in GB>,
  "RequiresStorageAuthentication": <Authentication requirement true or false>,
  "UserHasStoragePassword": <User has created password for this StorageVault true or false>
  "Company": {
    "Id": "<GUID Identifier of the Company>",
    "Name": "<Name of the Company>"
  },
  "UserStoragePasswordStatus": 0,
  "Default": <Denotes if this StorageVault is the default true or false>,
  "IrmRmsOption": <IRM feature related flag options>,
  "ConsumedGb": <Consumed Storage in GB>,
  "ProtocolVersion": 1.0
}
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
ID	GUID ID associated with the StorageVault endpoint	GUID	
Name	User defined name of the StorageVault.	String	
Description	Administrator's description of the StorageVault.	String	

Field	Description	Data Type	Valid Values
CompanyId	GUID that identifies the current user's company.	GUID	
Active	Flag that denotes whether the returned StorageVault is marked as active or not	Boolean	True or false
Urls	List of URLs that are associated with the returned StorageVault.	JSON Object	
Auths	List of datastore access keys that are associated with the returned StorageVault	JSON Object	
Url	URL that is associated and defined as an upload/download identifier of the returned StorageVault.	String	Standard URL format
Version	Currently unused	Integer	Always returns 0
SizeGb	Size in GB of the allocated storage available on the returned StorageVault.	Integer	Positive integer value ≥ 0
RequiresStorageAuthentication	Denotes if there is an authentication requirement.	Boolean	True or false

Field	Description	Data Type	Valid Values
UserHasStoragePassword	Denotes if specified user has stored a password for the returned StorageVault, only used StorageVault authentication feature is enabled.	Boolean	True or false
Company	Company information	JSON Object	
Company.Id	GUID that identifies the current user's company	GUID	
Company.Name	Name of the current user's company	String	
UserStoragePasswordStatus		Integer	Positive integer ≥ 0
Default	Denotes if this StorageVault is the default	Boolean	True or false
IrmRmsOption	IRM feature related flag options	Integer	Positive integer ≥ 0
ConsumedGb	Consumed Storage in GB	Integer	Positive integer ≥ 0
ProtocolVersion	Future use	Float	

Response Error Details

HTTP Code	Error Code	Description
-----------	------------	-------------

400	(Bad request) – Invalid input	
403	(Forbidden) - The caller is attempting to obtain StorageVault information not as an administrator	
403	(Forbidden) - user Not Authorized For Action	Only admin users can perform this request
403	(Forbidden) - Request Came From An Unapproved IP Address	When restriction by IP is enabled and that IP is not in the list of allowed IPs.
404	(Not found) - Storage Endpoint Not Found or Owner Does Not Exist or Invalid Storage Endpoint ID	

Syncpoints POST

Creates one or more syncpoints for a specific user (owner). The response list only contains those syncpoints that were successfully created. If an error occurred during the user creation process, a non-200 HTTP status code is returned.

Resource URL

<https://api.syncplicity.com/syncpoint/syncpoints.svc/>

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample JSON Input

```
[
  {
    "Type": <Syncpoint Type>,
    "Name": "<User Defined Name>",
    "Mapped": <Map to a local machine true or false>,
    "Path": "<Local File System Path>",
    "DownloadEnabled": <Allow Downloads true or false>,
    "UploadEnabled": <Allow Uploads true or false>,
    "StorageEndpointId": "<Storage Endpoint GUID>"
  }
]
```

Sample JSON Input Definitions

Param Name	Required	Description	Data Type	Valid Values
Type	Yes	Type of the syncpoint to be created. There are predefined syncpoint types for system folders (1-5). Only one system folder of each type (1-5) can be created. Attempts to create more than one system folder (1-5) will not succeed if the folder already exists. Custom syncpoints represent any other non-system folder and should be of type 6.	Integer Enum	1 - MyDocuments (predefined syncpoint for documents) 2 - MyMusic (predefined syncpoint for music) 3 - MyPictures (predefined syncpoint for pictures) 4 - Desktop (predefined Desktop syncpoint) 5 - Favorites (predefined Favorites syncpoint) 6 - Custom (predefined syncpoint for custom data) 8 - SyncDrive (predefined syncpoint for mapped to SyncDrive)
Name	Yes	New syncpoint's name	String	Non-empty string limited to 50 characters
Mapped	Yes	Whether the syncpoint is mapped to a machine or not	Boolean	True or false

Param Name	Required	Description	Data Type	Valid Values
Path	Yes	Path to the local file system directory of the syncpoint	String	Valid directory name, windows file system separator should be escaped with double back slash(\\), e.g. C:\\Users.
DownloadEnabled	Yes	Whether the syncpoint allows downloads	Boolean	True or false
UploadEnabled	Yes	Whether the syncpoint allows uploads	Boolean	True or false
StorageEndpointID	Yes	Unique ID associated with an existing storage endpoint.	GUID	If the field is empty, the owner's default storage endpoint will be used.

Sample JSON Response

```
[
  {
    "ID": "<Syncpoint ID>",
    "Type": 6,
    "Name": "<Syncpoint Name>",
    "Owner": {
      "ID": "<Owner's GUID>",
      "EmailAddress": "<Owner's Email>",
      "FirstName": "<Owner's First Name>",
      "LastName": "<Owner's Last Name>"
    },
    "Permission": 1,
    "StorageEndpointID": "<Storage Endpoint ID>",
    "Parent": {
      "ID": "<Parent Syncpoint ID>",
    },
    "RootFolderID": "<Root Folder ID>"
    "Mappings": [
      "Machine": {
        "ID": "<Owner Machine's GUID>",
        "Path": "<Owner Machine's Local Path to the Syncpoint Folder>"
      }
    ],
    "RemoteWipe": false,
    "Participants": [
      "user": [
        {
          "EmailAddress": "<Participant 1 Email>"
        },
        ...
      ]
    ]
  }
]
```

```

    ],
    "Group": [
      {
        "ID" : "<Group ID>",
        "Permission": 3
      },
      ...
    ]
  ],
  "SharingInviteNote": "A new folder <Syncpoint Name>
    has been shared with you.",
  "DownloadEnabled": true,
  "UploadEnabled": true,
  "ServerID": <Server ID>,
  "ServerStatus": 1
}
]

```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
ID	ID associated with the created syncpoint	Integer	
Type	Type of the syncpoint to be created. There are predefined syncpoint types for system folders (1-5). Only one system folder of each type (1-5) can be created. Attempts to create more than one system folder (1-5) will not succeed if the folder already exists. Custom syncpoints represent any other non-system folder and should be of type 6.	Integer Enum	1 - MyDocuments (predefined syncpoint for documents) 2 - MyMusic (predefined syncpoint for music) 3 - MyPictures (predefined syncpoint for pictures) 4 - Desktop (predefined Desktop syncpoint) 5 - Favorites (predefined Favorites syncpoint) 6 - Custom (predefined syncpoint for custom data) 8 - SyncDrive (predefined syncpoint for mapped to SyncDrive)
Name	New syncpoint's name	String	Non-empty string limited to 50 characters
Owner	Syncpoint's owner information	JSON Object	

Field	Description	Data Type	Valid Values
Owner.ID	Syncpoint's owner GUID	GUID	
Owner.EmailAddress	Yes if Owner object is defined, otherwise no	Email address of the syncpoint owner	Standard email format limited to 254 characters. Email should belong to user in the current company
Owner.FirstName	First name of the syncpoint owner	String	
Owner.LastName	Last name of the syncpoint owner	String	
Permission	Current user's permission for this syncpoint	Integer	0 - None (No permission to access the folder) 1- ReadWrite (Read/write permission or the owner of the folder) 2 - Contributor (Currently Ignored) 3 - ReadOnly (user can only download files from the Virtual Folder)
StorageEndpointID	Unique ID associated with an existing storage endpoint	GUID	If field is empty the owner's default storage endpoint will be used.
Parent	Parent syncpoint. Used for defining nested syncpoints, not a typical use-case	JSON Object	
Parent.ID	ID of parent syncpoint	Integer	Positive integer value > 0
RootFolderID	ID of the root folder	Long	Positive long value > 0

Field	Description	Data Type	Valid Values
Mappings	Mapping between a user's Syncplicity client and the syncpoint being created	JSON Array[1] (See caveat below)	Note: Although defined as a JSON array, currently the system will accept only one mapping on the POST call
Mappings.Machine	Mapped machine's information	JSON Object	
Mappings.Machine.ID	Unique ID associated with target machine	GUID	
Mappings.Machine.Path	Path of standard format which starts with '/' for Unix systems. Limited to 256 characters	String	Example c:\workdocuments or /usr/local/documents
RemoteWipe	When true, this request will trigger a remote wipe on the specified owner's machine	Boolean	true to trigger a remote wipe, false otherwise
Participants	Array of syncpoint sharing participants	JSON Array	
Participants.user	User to be added as a participant of the new syncpoint	JSON Array[] EmailAddress	
Participants.user.Email Address	Email address of the syncpoint user participant	Email	Must be a valid user in the company. User is ignored if the user is participating in a parent folder already
Participants.Group	Group participant information	JSON Array[] Group information	
Participants.Group.ID	Unique ID associated with the group participant	GUID	

Field	Description	Data Type	Valid Values
Participants.Permission	User or group permission for this syncpoint	Integer	0 - None (No permission to access the folder) 1 - ReadWrite (Read/write permission or the owner of the folder) 2 - Contributor (Currently Ignored) 3 - ReadOnly (user can only download files from Virtual Folder)
SharingInviteNote	A custom message used in the invite email sent to participants	Boolean	
DownloadEnabled	True if download or synchronization is enabled for this syncpoint	Boolean	
UploadEnabled	True if this syncpoint has backup/upload enabled	String	
ServerID	ID of server associated with the syncpoint	Integer	
ServerStatus	No	Status of syncpoint server	Integer 0 - Unknown (Unknown status) 1 - Online (Online status) 2 - Offline (Offline status)

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request) – Invalid syncpoints list	
403	Forbidden) - Not A Business Admin - Caller is attempting to create syncpoints for other users but is not a company administrator	
403	(Forbidden) - user Not Authorized For Action	

Syncpoints GET (Optional)

Retrieves the syncpoint definitions (top-level folders) for a specific user or for the current user.

Resource URL

```
https://api.syncplicity.com/syncpoint/syncpoints.svc/?user={USER_GUID}&include={include}
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
includeType={includeType}	No	If specified, returns the syncpoint definitions of the specified type. Note: Syncpoints of type 8 - SyncDrive are returned only when this parameter is specified.	String	One or several values, separated by comma. Possible Values: 1 - MyDocuments (predefined syncpoint for documents) 2 - MyMusic (predefined syncpoint for music) 3 - MyPictures (predefined syncpoint for pictures) 4 - Desktop (predefined Desktop syncpoint) 5 - Favorites (predefined Favorites syncpoint) 6 - Custom (predefined syncpoint for custom data) 8 - SyncDrive (predefined syncpoint for mapped to SyncDrive)
user={USER_GUID}	No	If specified, retrieves the syncpoint definitions of the specified user; otherwise, the current user of the application is used.	GUID	Application must be working on behalf of an administrator when specifying USER_GUID for a different user
include={include}	No	If specified, make service return additional	String	One or several values, separated by comma.

information about specified objects

Possible values:

Participants – include Participants list (shared users, invited users and shared groups);

Inviter – include user who invited current user to this syncpoint;

remoteWipe – include remote wiped syncpoints;

children – include child syncpoints;

lastnewsfeedevent – fill LastEvent property with the last news feed, or, if it is NULL – latest newsfeed archive event

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token

Name	Value	Description
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample JSON Response

```
[
  {
    "ID": "<Syncpoint ID>",
    "Type": 6,
    "Name": "<Syncpoint Name>",
    "RootFolderId": "<Root folder ID>",
    "Mapped": false,
    "DownloadEnabled": false,
    "UploadEnabled": false,
    "Shared": true,
    "Owner": {
      "ID": "<Owner's GUID>",
      "EmailAddress": "<Owner's Email>",
      "FirstName": "<Owner's First Name>",
      "LastName": "<Owner's Last Name>",
      "Company": {
        "Id": "<Company's Id>",
        "Name": "<Company's Name>"
      }
    },
    "Permission": 1,
    "StorageEndpointID": "<Storage Endpoint ID>",
    "Parent": {
      "ID": "<Parent Syncpoint ID>",
    },
    "Mappings": [
      "Machine": {
        "ID": "<Owner Machine's GUID>",
        "Path": "<Owner Machine's Local Path to the Syncpoint Folder>"
      }
    ],
    "RemoteWipe": false,
  }, ...
]
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
ID	ID associated with the syncpoint	Integer	

Field	Description	Data Type	Valid Values
Type	Type of the syncpoint to be created. There are predefined syncpoint types for system folders (1-5). Only one system folder of each type (1-5) can be created. Attempts to create more than one system folder (1-5) will not succeed if the folder already exists. Custom syncpoints represent any other non-system folder and should be of type 6.	Integer Enum	1 - MyDocuments (predefined syncpoint for documents) 2 - MyMusic (predefined syncpoint for music) 3 - MyPictures (predefined syncpoint for pictures) 4 - Desktop (predefined Desktop syncpoint) 5 - Favorites (predefined Favorites syncpoint) 6 - Custom (predefined syncpoint for custom data) 8 - SyncDrive (predefined syncpoint for mapped to SyncDrive)
Name	Syncpoint name	String	Non-empty string limited to 50 characters
RootFolderId	ID of root folder in the syncpoint. Could be used for retrieving content of syncpoint.	Long	
Mapped	Whether syncpoint has been mapped to any devices	Boolean	Note: kept for backwards compatibility - syncPoint. Mappings is a more modern approach
DownloadEnabled	True if download/synchronization is enabled for this syncpoint	Boolean	
UploadEnabled	True if this syncpoint has backup/upload enabled	Boolean	

Field	Description	Data Type	Valid Values
Shared	Whether syncpoint is shared with anybody	Boolean	
Owner	Syncpoint's owner information	JSON Object	
Owner.ID	Syncpoint's owner GUID	GUID	
Owner.EmailAddress	Yes if Owner object is defined, otherwise no	Email address of the syncpoint owner	Email Standard email format limited to 254 characters. Email should belong to user in the current company
Owner.FirstName	First name of the syncpoint owner	String	
Owner.LastName	Last name of the syncpoint owner	String	
Permission	Current user's permission for this syncpoint	Integer	0 - None (No permission to access the folder) 1- ReadWrite (Read/write permission or the owner of the folder) 2 - Contributor (Currently Ignored) 3 - ReadOnly (user can only download files from the Virtual Folder)
Participants	If included, list of all participants (shared users, invited users and shared groups)	JSON object	
Policy	Information about policies applied to syncpoint, for example	JSON object	

Field	Description	Data Type	Valid Values
	ShareLinkPolicy, StoragePasswordPolicy, etc		
StorageEndpointID	Unique ID associated with an existing storage endpoint	GUID	If the field is empty, the owner's default storage endpoint will be used.
Parent	Parent syncpoint. Used for defining nested syncpoints, not a typical use-case	JSON Object	
Parent.ID	ID of the parent syncpoint	Integer	Positive integer value > 0
Mappings	Mapping between user's Syncplicity client and the syncpoint being created	JSON	
Mappings.Machine	Mapped machine's information	JSON Object	
Mappings.Machine.ID	Unique ID associated with the target machine	GUID	
Mappings.Machine.Path	Path of standard format which starts with '/' for Unix systems. Limited to 256 characters	String	Example c:\workdocuments or /usr/local/documents
RemoteWipe	When true, this request will trigger a remote wipe on the specified owner's machine	Boolean	true to trigger a remote wipe, false otherwise

Field	Description	Data Type	Valid Values
Mappings.DownloadEnabled	True if download/synchronization is enabled for this syncpoint	Boolean	
Mappings.UploadEnabled	True if this syncpoint has backup/upload enabled	String	
EventServerId	ID of event server associated with the syncpoint	GUID	
ServiceState.Status	Status of the server	Integer	0 - Unknown (Unknown status) 1 - Online (Online status) 2 - Offline (Offline status)
LastEvent	Information about last newsfeed event with this syncpoint. Contains event Id by which additional info could be requested from newsfeed.svc	JSON object	
SyncpointSize	Size of syncpoint in bytes	Integer	
ResharingPermission	Defines resharing permission – who can reshare syncpoint	Integer enum	None = 0 AllowAll = 1 AllowIntended = 2
NotifyAboutResharing	Enables notifications to participants when resharing performed	Boolean	

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request) - invalid Email Format	The format of the supplied email address is bad
401	(Unauthorized) - Not Authorized	The token is invalid, the header is missing, or the user is inactive or disabled
403	(Forbidden) - Not A Business Admin	The current user is not an administrator and the "user" query parameter is specified
403	(Forbidden) - user Not Authorized For Action	
404	(Not Found) - user Not Found	The "user" query parameter is specified, but the user does not exist

Syncpoint GET

Retrieves the syncpoint definition of a specific endpoint.

Resource URL

`https://api.syncplicity.com/syncpoint/syncpoint.svc/{SYNCPOINT_ID}`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	No	Integer ID of the syncpoint	Integer	Positive number > 0

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>

AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample JSON Response

```
{
  "ID": <Syncpoint ID>,
  "Type": 6,
  "Name": "<Syncpoint Name>",
  "Owner": {
    "ID": "<Owner's GUID>",
    "EmailAddress": "<Owner's Email>",
    "FirstName": "<Owner's First Name>",
    "LastName": "<Owner's Last Name>"
  },
  "Permission": 1,
  "StorageEndpointID": "<Storage Endpoint ID>",
  "Parent": {
    "ID" : "<Parent Syncpoint ID>",
  },
  "RootFolderID": <Root Folder ID>
  "Mappings": [
    "Machine": {
      "ID" : "<Owner Machine's GUID>",
      "Path": "<Owner Machine's Local Path to the Syncpoint Folder>"
    }
  ],
  "RemoteWipe": false,
  "DownloadEnabled": true,
  "UploadEnabled": true,
  "ServerID": 3,
  "ServerStatus": 1
}
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
ID	ID associated with the syncpoint	Integer	
Type	Type of the syncpoint to be created. There are predefined syncpoint types for system folders (1-5). Only one system folder of each type (1-5) can be created. Attempts to create more than one system folder (1-5) will not succeed if the folder already exists. Custom syncpoints represent any other non-system folder and should be of type 6.	Integer Enum	<p>1 - MyDocuments (predefined syncpoint for documents)</p> <p>2 - MyMusic (predefined syncpoint for music)</p> <p>3 - MyPictures (predefined syncpoint for pictures)</p> <p>4 - Desktop (predefined Desktop syncpoint)</p> <p>5 - Favorites (predefined Favorites syncpoint)</p> <p>6 - Custom (predefined syncpoint for custom data)</p> <p>8 - SyncDrive (predefined syncpoint for mapped to SyncDrive)</p>
Name	New syncpoint's name	String	Non-empty string limited to 50 characters
Owner	Syncpoint's owner information	JSON Object	
Owner.ID	Syncpoint's owner GUID	GUID	
Owner.EmailAddress	Yes if Owner object is defined, otherwise no	Email address of the syncpoint owner	Standard email format limited to 254 characters. Email should belong to a user in the current company

Field	Description	Data Type	Valid Values
Owner.FirstName	First name of the syncpoint owner	String	
Owner.LastName	Last name of the syncpoint owner	String	
Permission	Current user's permission for this syncpoint	Integer	<p>0 - None (No permission to access the folder)</p> <p>1 - ReadWrite (Read/write permission or the owner of the folder)</p> <p>2 - Contributor (Currently Ignored)</p> <p>3 - ReadOnly (user can only download files from the Virtual Folder)</p>
StorageEndpointID	Unique ID associated with an existing storage endpoint	GUID	If the field is empty, the owner's default storage endpoint will be used.
Parent	Parent syncpoint. Used for defining nested syncpoints, not a typical use-case	JSON Object	
Parent.ID	ID of parent syncpoint	Integer	Positive integer value > 0
RootFolderID	ID of root folder	Long	Positive long value > 0
Mappings	Mapping between a user's Syncplicity client and the syncpoint being created	JSON Array[1] (See caveat below)	Note: although defined as a JSON array, currently the system will accept only one mapping on the POST call
Mappings.Machine	Mapped machine's information	JSON Object	

Field	Description	Data Type	Valid Values
Mappings.Machine.ID	Unique ID associated with target machine	GUID	
Mappings.Machine.Path	Path of standard format which starts with '/' for Unix systems. Limited to 256 characters	String	Example c:\workdocuments or /usr/local/documents
DownloadEnabled	True if download / synchronization is enabled for this syncpoint	Boolean	
UploadEnabled	True if this syncpoint has backup/upload enabled	String	
ServerStatus	No	Status of syncpoint server	Integer 0 - Unknown (Unknown status) 1 - Online (Online status) 2 - Offline (Offline status)

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request) - invalid Email Format	Format of the supplied email address is wrong
401	(Unauthorized) - Not Authorized	The token is invalid, header is missing, or user is inactive or disabled
403	(Forbidden) - Not A Business Admin	Current user is not an administrator and the "user" query parameter is specified

HTTP Code	Error Code	Description
403	(Forbidden) - user Not Authorized For Action	
404	(Not Found) - user Not Found	The "user" query parameter is specified, but the user does not exist

Syncpoint PUT

Updates a Syncpoint specified by the ID. This call can be used to update properties of the Syncpoint, as well as map or un-map a Syncpoint from an endpoint. Included properties of the input are updated if the values don't match existing values stored on the server, otherwise inputs are ignored if unchanged. Therefore, only changed values need be sent in the JSON input.

Resource URL

https://api.syncplicity.com/syncpoint/syncpoint.svc/{SYNCPOINT_ID}

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal

Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Sample JSON Input

```
{
  "ID": "<Syncpoint ID>",
  "Type": 6,
  "Name": "<Syncpoint Name>",
  "Owner": {
    "ID": "<Owner's GUID>",
    "EmailAddress": "<Owner's Email>",
    "FirstName": "<Owner's First Name>",
    "LastName": "<Owner's Last Name>"
  },
  "StorageEndpointId": "<Storage Endpoint ID>",
  "RootFolderId": "<Root Folder ID>",
  "Mappings": [{
    "Mapping": {
      "SyncPointId": "<Syncpoint ID>",
      "Machine": {
        "ID": "<Owner Machine's ID>"
      },
      "Path": "C:\\Users\\user1\\Syncplicity Folders\\syncpoint1\\",
      "Mapped": "true",
      "DownloadEnabled": "true",
      "UploadEnabled": "true"
    }
  }]
}
```

Sample JSON Input Definitions

Param Name	Required	Description	Data Type	Valid Values
Type	Yes	Type of the syncpoint to be created. There are predefined syncpoint types for system folders (1-5). Only one system folder of each type (1-5) can be created. Attempts to create more than one system folder (1-5) will not succeed if the folder already exists. Custom syncpoints represent any other non-system folder and should be of type 6.	Integer Enum	<p>1 - MyDocuments (predefined syncpoint for documents)</p> <p>2 - MyMusic (predefined syncpoint for music)</p> <p>3 - MyPictures (predefined syncpoint for pictures)</p> <p>4 - Desktop (predefined Desktop syncpoint)</p> <p>5 - Favorites (predefined Favorites syncpoint)</p> <p>6 - Custom (predefined syncpoint for custom data)</p> <p>8 - SyncDrive (predefined syncpoint for mapped to SyncDrive)</p> <p>NOTE: The type of an existing syncpoint property should be specified but cannot be changed when performing a Syncpoint PUT.</p>
Name	Yes	New syncpoint's name	String	Non-empty string limited to 50 characters
Owner	No	Syncpoint's owner information	JSON Object	Note: if not part of the request, the current user will be used as the owner of the new syncpoint
Owner.EmailAddress	Yes if Owner object is defined, otherwise no	Email address of the syncpoint owner	Email	Standard email format limited to 254 characters. Email should belong to user in the current company

Param Name	Required	Description	Data Type	Valid Values
StorageEndpointID	No	Unique ID associated with an existing storage endpoint	GUID	If the field is empty the owner's default storage endpoint will be used. Note: The StorageEndPointID of an existing syncpoint cannot be changed when performing a Syncpoint PUT.
RootFolderID	Yes if Parent object is defined, otherwise no	ID of the root folder	Long	Positive long value > 0. Note: The RootFolderID of an existing syncpoint cannot be changed when performing a Syncpoint PUT.
Mappings	No	Mapping between a user's Syncplicity client and the syncpoint being created	JSON Array[1] (See caveat below)	Note: although defined as a JSON array, currently the system will accept only one mapping on the POST call
Mappings.Machine	Yes if Mappings are defined, otherwise no	Mapped machine's information	JSON Object	
Mappings.Machine.ID	Yes if Mappings.Machine object is defined, otherwise no	Unique ID associated with the target machine	GUID	
Mappings.Machine.Path	Yes if Mappings.Machine object is defined, otherwise no	Path of standard format which starts with '/' for Unix systems. Limited to 256 characters	String	Example c:\workdocuments or /usr/local/documents

Sample JSON Response

```
{
  "ID": "<Syncpoint ID>",
  "Type": 6,
  "Name": "<Syncpoint Name>",
  "Mapped": "true",
  "Path": "C:\\Users\\user1\\Syncplicity Folders\\syncpoint1\\",
  "DownloadEnabled": "true",
  "UploadEnabled": "true",
  "Owner": {
    "ID": "<Owner's GUID>",
    "EmailAddress": "<Owner's Email>",
    "FirstName": "<Owner's First Name>",
    "LastName": "<Owner's Last Name>"
  },
  "Mappings": {
    "Mapping": {
      "SyncPointId": "<Syncpoint ID>",
      "Machine": {
        "ID": "<Owner Machine's ID>"
      },
      "Path": "C:\\Users\\user1\\Syncplicity Folders\\syncpoint1\\",
      "Mapped": "true",
      "DownloadEnabled": "true",
      "UploadEnabled": "true"
    }
  },
  "ServiceState": {
    "Status": 1
  }
}
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
ID	ID associated with the created syncpoint	Integer	
Type	Type of the syncpoint to be created. There are predefined syncpoint types for system folders (1-5). Only one system folder of each type (1-5) can be created. Attempts to create more than one system folder (1-5) will not succeed if the	Integer Enum	1 - MyDocuments (predefined syncpoint for documents) 2 - MyMusic (predefined syncpoint for music) 3 - MyPictures (predefined syncpoint for pictures) 4 - Desktop (predefined Desktop syncpoint) 5 - Favorites (predefined Favorites syncpoint) 6 - Custom (predefined syncpoint for custom data)

Field	Description	Data Type	Valid Values
	folder already exists. Custom syncpoints represent any other non-system folder and should be of type 6.		8 – SyncDrive (predefined syncpoint for mapped to SyncDrive)
Name	New syncpoint's name	String	Non-empty string limited to 50 characters
Owner	Syncpoint's owner information	JSON Object	
Owner.ID	Syncpoint's owner GUID	GUID	
Owner.EmailAddress	Yes if Owner object is defined, otherwise no	Email address of syncpoint owner	Standard email format limited to 254 characters. Email should belong to user in the current company
Owner.FirstName	First name of the syncpoint owner	String	
Owner.LastName	Last name of the syncpoint owner	String	
StorageEndpointID	Unique ID associated with an existing storage endpoint	GUID	If the field is empty the owner's default storage endpoint will be used.
RootFolderID	ID of the root folder	Long	Positive long value > 0
Mappings	Mapping between a user's Syncplicity client and the syncpoint being created	JSON Array[1] (See caveat below)	Note: although defined as a JSON array, currently the system will accept only one mapping on the POST call
Mappings.Machine	Mapped machine's information	JSON Object	

Field	Description	Data Type	Valid Values
Mappings.Machine.ID	Unique ID associated with the target machine	GUID	
Mappings.Machine.Path	Path of standard format which starts with '/' for Unix systems. Limited to 256 characters	String	Example c:\workdocuments or /usr/local/documents
DownloadEnabled	True if download/synchronization is enabled for this syncpoint	Boolean	
UploadEnabled	True if this syncpoint has backup/upload enabled	String	
ServerStatus	No	Status of syncpoint server	Integer 0 - Unknown (Unknown status) 1 - Online (Online status) 2 - Offline (Offline status)

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request) - Invalid Folder Name	Invalid Folder Name
400	(Bad request) - Owner user does not Exist	Owner user does not exist
400	(Bad request) - Invalid Machine ID	Invalid Machine ID
403	(Forbidden) - Syncpoint Rename Forbidden	Syncpoint Rename Forbidden

HTTP Code	Error Code	Description
403	(Forbidden) - Folder Rename not allowed	Folder Rename is not allowed
403	(Forbidden) - Syncpoint Change Owner Forbidden	Syncpoint Change Owner Forbidden
403	(Forbidden) - Only Company users Allowed	Only Company users allowed
403	(Forbidden) - Sub-folder Sharing Change Owner Forbidden	Sub-folder Sharing Change Owner Forbidden

Syncpoint DELETE (Optional)

Removes an existing Syncpoint from the Syncplicity account.

Resource URL

`https://api.syncplicity.com/syncpoint/syncpoint.svc/{SYNCPOINT_ID}`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	Unique ID associated with the syncpoint	Integer	

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Response Error Details

HTTP Code	Error Code	Description
403	(Forbidden) - user Not Authorized For Action	user is Not Authorized to perform this Action
403	(Forbidden) - Virtual Folder Is Shared Via The Group Permissions	Virtual Folder Is Shared Via The Group Permissions

Folders POST

Creates one or more subfolders within a given syncpoint and specified parent folder.

Resource URL

https://api.syncplicity.com/sync/folder_folders.svc/{SYNCPOINT_ID}/folder/{FOLDER_ID}/folders

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. A private syncpoint will not be able to be referenced by this API
{FOLDER_ID}	Yes	ID associated with the folder within the syncpoint. NOTE: If the folder is being created in the root of a syncpoint, the Syncpoint GET API call must be used to get the RootFolderID and that value should be used here.	Long	Positive long value > 0

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Sample JSON Input

```
[
  {
    "Name": "Sub-folder",
    "Status": 1
  },
  {
    "Name": "Sub-folder 2",
    "Status": 1
  },
  ...
]
```

Sample JSON Input Definitions

Param Name	Required	Description	Data Type	Valid Values
Name	Yes	Name of folder to be created	String	
Status	Yes	The folder status	Integer Enum	0 - None 1 - Added 4 - Removed 5 - ConfirmedRemoved

Sample JSON Response

```
[
  {
    "SyncpointId": <Syncpoint ID>,
    "FolderId": <Folder ID>,
    "VirtualPath": "\Sub-folder\",
    "Name": "Sub-folder",
    "Status": 1
  },
  {
    "SyncpointId": <Syncpoint ID>,
    "FolderId": <Folder ID>,
    "VirtualPath": "\Sub-folder 2\",
    "Name": "Sub-folder 2",
    "Status": 1
  },
  ...
]
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
SyncpointId	ID associated with the folder syncpoint	Integer	
FolderId	ID associated with the created folder	Long	
Name	Name of the created folder	String	

Field	Description	Data Type	Valid Values
VirtualPath	Virtual path to the created folder (path from the syncpoint's point of view)	String	
Status	The folder status	Integer Enum	0 - None 1 - Added 4 - Removed 5 - ConfirmedRemoved

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request) - Invalid Folder ID	Invalid Folder ID
400	(Bad request) - Different Statuses Are Not Allowed	Different Statuses Are Not Allowed
400	(Bad request) - Bad Request	Bad Request
400	(Bad request) - Empty Folder Names Not Allowed	Empty Folder Names Not Allowed
400	(Bad request) - Invalid Status	Invalid Status
403	(Forbidden) - user Not Authorized For Action	User is Not Authorized to perform this Action

Folders GET

Retrieve the list of files/folders contained within the specified folder within a given syncpoint.

Resource URL

```
https://api.syncplicity.com/sync/folder_folders.svc/{SYNCPOINT_ID}/folder/{FOLDER_ID}/folders?limit={MAX_ITEMS}&last_folder_id={LAST_FOLDER_ID}
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. A private syncpoint will not be able to be referenced by this API
{FOLDER_ID}	Yes	ID associated with the parent folder within the syncpoint	Long	Positive long value > 0
{MAX_ITEMS}	No	Maximum number of items to be loaded	Integer	Positive integer value > 0
{LAST_FOLDER_ID}	No	Last loaded folder ID	Long	Positive long value > 0

URL Query Parameters

Name	Value	Description
uhid={user_email_hash}	Yes	If specified, retrieves the storage endpoint (StorageVault) definitions of the specified user; otherwise, the current user of the application is used. String User email hash is calculated using the user's email and encoding it using the HMAC MD5 algorithm with "6adebb9f-21f9-49d8-95bf-b7007a208cd4" salt

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Sample JSON Response

```
{
  "SyncpointId": <Syncpoint ID>,
  "FolderId": <Folder ID>,
  "VirtualPath": "\Sub-Folder",
  "Name": "Sub-Folder",
  "Status": 1,
  "Files": [
    {
      "SyncpointId": <Syncpoint ID>,
      "FileId": <File ID>,
      "Filename": "file01.txt",
      "Length": 3473,
      "LastWriteTimeUtc": "2015-02-13T08:43:07.6530000Z",
      "Status": 1,
      "LatestVersionId": <Latest Version ID>,
      "ThumbnailUrl": "<thumbnail url>",
    },
    ...
  ],
  "Folders": [
    {
      "SyncpointId": <Syncpoint ID>,
      "FolderId": <Folder ID>,
      "VirtualPath": "\Sub-Folder\",
      "Name": "Sub-folder 2",
      "Status": 1
    },
    ...
  ]
}
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
SyncpointId	ID associated with the folder syncpoint	Integer	
FolderId	ID associated with the target folder	Long	
VirtualPath	Virtual path to the folder (path from the syncpoint's point of view)	String	

Field	Description	Data Type	Valid Values
Name	Name of the specified folder	String	Empty for the root folders (the syncpoint itself)
Status	Folder status	Integer Enum	0 - None 1 - Added 4 - Removed 5 - ConfirmedRemoved
Files	List of files in the specified folder (may be empty)	JSON Object	Will only be returned if {INCLUDE} is set accordingly
Files.SyncpointId	ID associated with the file syncpoint	Integer	
Files.FileId	file ID	Integer	
Files.FileName	Name of the file	String	
Files.Length	Length of the file	Integer	
Files.LastWriteTimeUtc	Last write time	String (UTC)	
Files.Status	The file status	Integer Enum	0 - None 1 - Added 4 - Removed 5 - ConfirmedRemoved
Files.LatestVersionId	Latest file version ID	Integer	
Files.ThumbnailUrl	Thumbnail URL to the file	String (URL)	
Folders	Sub-folders which are located within the target folder	JSON Object	Will only be returned if {INCLUDE} is set accordingly
Folders.SyncpointId	ID associated with the sub-folder syncpoint	Integer	

Field	Description	Data Type	Valid Values
Folders.FolderId	ID associated with the sub-folder	Long	
Folders.VirtualPath	Virtual path to the folder (path from the syncpoint's point of view)	String	
Folders.Name	Name of the sub-folder	String	
Folders.Status	The folder status	Integer Enum	0 - None 1 - Added 4 - Removed 5 - ConfirmedRemoved

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request) - Invalid Folder ID	Invalid Folder ID.

Folder DELETE (Optional)

Delete a given folder and all of its contents.

Resource URL

```
https://api.syncplicity.com/sync/folder.svc/{SYNCPOINT_ID}/folder/{FOLDER_ID}?modifier={MODIFIER}
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. A private syncpoint will not be able to be referenced by this API.
{FOLDER_ID}	Yes	ID associated with the folder within the syncpoint	Long	Positive long value > 0

URL Query Parameters

Name	Description
modifier	{MODIFIER} can be one of "permanently" or <empty>. When the value is "permanently" this deletes the folder and contents permanently. When <empty> or not set, the folder and contents can be restored based on file retention policies set by the administrator.

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Upload File [saveFile POST]

The file upload is performed by invoking a multipart/form-data POST to the <storage endpoint url>/saveFile.php (from Storage Endpoint GET). The endpoint URL must be the storage endpoint that hosts the referenced syncpoint.

Files larger than 5 MB should be uploaded in 5 MB chunks as follows:

1. Upload the first 5 MB chunk of the file and be sure to **not** include the fileDone body part (unless the file is less than 5 MB in size).
2. The successful HTTP response will be "Resume Incomplete" and the HTTP response code will be "308 Redirection" if this chunk is not the last one, and the HTTP response will be "Stored" and the HTTP response code will be will be "200 OK" if this is the last chunk.
3. If there is more than one chunk, Read the **Etag** HTTP header returned in the response.
4. You can also read the Range HTTP header returned in the response which includes the byte range of the successful chunk upload.
5. Upload the second 5 MB file chunk and be sure to include the **If-Match** HTTP header with the value set to the ETag value from step 3 above. ONLY include the fileDone body part if this is the last chunk.
6. The successful HTTP response will be "Resume Incomplete" and the HTTP response code will be "308 Redirection" if this chunk is not the last one, and the HTTP response will be "Stored" and the HTTP response code will be will be "200 OK" if this is the last chunk.
7. You can also read the Range HTTP header returned in the response which includes the byte range of all successful chunk uploads.
8. If there are more chunks then upload the next one and be sure to include the **If-Match** HTTP header with the value set to the ETag value from step 3 above. ONLY include the fileDone body part if this is the last chunk.
9. The successful HTTP response will be "Resume Incomplete" and the HTTP response code will be "308 Redirection" if this chunk is not the last one, and the HTTP response will be "Stored" and the HTTP response code will be will be "200 OK" if this is the last chunk.
10. You can also read the Range HTTP header returned in the response which includes the byte range of all successful chunk uploads.
11. Repeat steps 8, 9 and 10 until the last chunk has been uploaded.

The multi-part body (see <http://tools.ietf.org/html/rfc2388>) of file upload POST requests must include:

- fileData: Binary contents of the file
- sha256: SHA-256 hash of the contents of the file
- sessionKey: Machine bearer token issued by the Syncplicity authentication service (see [Programming Flow](#), step 2). The 'Bearer ' part should not be included in this case.
- virtualFolderId: ID of the Syncplicity folder where the file is located
- creationTimeUtc: File creation date and time
- lastWriteTimeUtc: File updated date and time
- fileDone: ONLY included with the last chunk of a file. If the file size is less than 5 MB then this body part will be included with the first and only chunk.

Resource URL

```
<storage endpoint url>/saveFile.php?filepath={FILE_PATH}
```

HTTP Headers Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app
Authorization (required)	Bearer <token>	Machine bearer token (see Programming Flow , step 2)
Syncplicity-Storage-Authorization (required for SVA)		Storage token for a StorageVault configured with Authentication (see Setup Procedure , step 4)
Content-Type (required)	multipart/form-data; boundary=----- <upload time in ticks>	Content type of request with upload time in ticks

Name	Value	Description
If-Match (optional)	<ETag>	ETag of file upload session. It is returned from server after very first chunk is uploaded.
Content-Encoding	gzip	Content compression
Content-Range	*/ <startByte>-*/*	For non-chunked upload For chunked upload
Content-Encoding (optional)	gzip	Content compression

URL Query Parameters

Param Name	Required	Description	Data Type	Valid Values
filepath={FILE_PATH}	Yes	Virtual file path from syncpoint view. Includes file name. Example: %5Csubfolder1%5C1234.txt NOTE: Use %5C instead of a backslash (\).	String	

Form Data

Name	Value	Description
Content-Disposition: form-data; name="fileData"; filename="<file name>" Content-Transfer-Encoding: binary Content-Type: application/octet-stream	<file content>	<file name> - name of the uploading file <file stream> - file content
Content-Disposition: form-data; name="sha256"		SHA256 hash of file content

Name	Value	Description
Content-Disposition: form-data; name="virtualFolderId"	<int>	Virtual folder ID
Content-Disposition: form-data; name="creationTimeUtc"		File creation date time in ISO 8601 format
Content-Disposition: form-data; name="lastWriteTimeUtc"		File change date time in ISO 8601 format
Content-Disposition: form-data; name="fileDone"		Marker of last file chunk

Sample Form Data

```
Content-Disposition: form-data; name="fileData"; filename="1234.txt"
Content-Transfer-Encoding: binary
Content-Type: application/octet-stream
<file>
```

```
Content-Disposition: form-data; name="sha256"
688787d8ff144c502c7f5cffaafe2cc588d86079f9de88304c26b0cb99ce91c6
```

```
Content-Disposition: form-data; name="sessionKey"
Bearer Y1FzUQHURak2MTY4lcga9hxP9yPO
```

```
Content-Disposition: form-data; name="virtualFolderId"
4250509
```

```
Content-Disposition: form-data; name="creationTimeUtc"
2015-10-27T23:07:54.3268208Z
```

```
Content-Disposition: form-data; name="lastWriteTimeUtc"
2015-10-27T23:08:02.4128366Z
```

```
Content-Disposition: form-data; name="fileDone"
```

Sample Response

Stored

Response Definitions

Response body may contain the following values:

Stored

File is successfully stored

Already Stored	File with the same content was already stored
----------------	---

Resume Incomplete	File chunk has been successfully stored, but the final chunk has not yet been received
-------------------	--

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request)	Bad request
401	(Not authorized)	Invalid authorization or storage token

Download File [retrieveFile GET] (Optional)

The file download is performed by invoking GET request against the <storage endpoint url>/retrieveFile.php (from Storage Endpoint GET). The endpoint url must be the storage endpoint that hosts the referenced syncpoint.

Resource URL

<storage endpoint url>/retrieveFile.php?vToken={V_TOKEN}

URL Query Parameters

Param Name	Required	Description	Data Type	Valid Values
{V_TOKEN}	Yes	File identifier in format <syncpoint id>-<last file version id> Example: V_TOKEN: 9999999-11111111	String	

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Machine bearer token (see Programming Flow, step 2)

Name	Value	Description
Syncplicity-Storage-Authorization (required for SVA)		Storage token for a StorageVault configured with Authentication (see Setup Procedure, step 4)
Range (optional)	<offset>-	<offset> is a start byte of file to be downloaded in case download resume is required. NOTE: The '-' after <offset> implies from offset to the end of the file.

Response

File stream of downloaded file.

Response Error Details

HTTP Code	Error Code	Description
400	(Bad request)	Bad request
401	(Not authorized)	Invalid authorization or storage token

Files GET

Gets a list of files within a given syncpoint, parent folder Id and ability to paginate.

Resource URL

```
https://api.syncplicity.com/sync/folder_files.svc/{SYNCPOINT_ID}/folder/{FOLDER_ID}/files?limit={MAX_ITEMS}&offset={LAST_FILE_ID}&include={include}
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. A private syncpoint will not be able to be referenced by this API
{FOLDER_ID}	Yes	ID associated with the folder within the syncpoint	Long	Positive long value > 0
{LIMIT}	No	Max. Number of items to be loaded per service call.	Integer	Positive integer value > 0
{LAST_FILE_ID}	No	Last loaded file Id	Long	Positive long value > 0
{INCLUDE}	No	Modifier to the API behavior.	String	Can be one of "active" and/or "delete", "active, delete" or <empty>. When the value is "active", only active files/folders will be returned. When the value is "delete", only deleted items will be returned. When the value is "active, delete", then both active and deleted files/folders are returned. When the

Param Name	Required	Description	Data Type	Valid Values
------------	----------	-------------	-----------	--------------

value is <empty> or not set, only the folder meta data is returned (no files/sub-folders)

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Sample JSON Response

```
[
  {
    "SyncpointId": <Syncpoint ID>,
    "FileId": <File ID>,
    "Filename": <File name>,
    "CreationTimeUtc": <File creation date>,
    "LastWriteTimeUtc": <File modification date>,
    "DateAddedUtc": <File upload date associated with ClientFileVersionId>,
    "Status": <File status>,
    "LatestVersionId": "<File version id>",
    "FolderId": "<Parent folder id>",
    "ClientFileVersionId": <>
  },
  {
    "SyncpointId": <Syncpoint ID>,
    "FileId": <File ID>,
    "Filename": <File name>,
    "CreationTimeUtc": <File creation date>,
    "LastWriteTimeUtc": <File modification date>,
    "DateAddedUtc": <File upload date associated with ClientFileVersionId>,
    "Status": <File status>,
    "LatestVersionId": "<File version id>",
    "FolderId": "<Parent folder id>",
    "ClientFileVersionId": <>
  },
  ...
]
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
SyncpointId	ID associated with the folder syncpoint	Integer	Positive integer value > 0
FileId	ID associated with the target file	Long	Positive long value > 0
Filename	Name of the file	String	Non-empty string
CreationTimeUtc	File creation date	DateTime	
LastWriteTimeUtc	File modification date	DateTime	Empty for the root folders (the syncpoint itself)

Field	Description	Data Type	Valid Values
DateAddedUtc	File upload date associated with ClientFileVersionId	DateTime	
Status	File status	Integer	0 - None 1 - Added 2 - Updated 3 - Removed 4 - Ignored 5 - ConfirmedRemoved
LatestVersionId	File last version ID	Long	Positive long value > 0
FolderId	Parent folder ID	Long	Positive long value > 0

File DELETE (Optional)

Delete a file by specifying the syncpoint ID and the file ID.

Resource URL

```
https://api.syncplicity.com/sync/file.svc/{SYNCPOINT_ID}/file/{FILE_ID}?modifier={MODIFIER}
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the file syncpoint	Integer	Positive integer value > 0
{FILE_ID}	Yes	IDs of the target file	Long	Positive long value > 0

URL Query Parameters

Name	Description
modifier	Set the value to "permanently" delete a file permanently

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>

Name	Value	Description
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Response Error Details

HTTP Code	Error Code	Description
400	Invalid SyncPoint ID	Invalid SyncPoint ID.
400	Invalid Version ID	Invalid Version ID.
400	Mismatched SyncPoint ID	Mismatched SyncPoint ID.
400	Purging Active File Not Allowed (only if modifier=permanently)	Purging Active File Not Allowed (only if modifier=permanently).
400	File Already Deleted	File Already Deleted.
403	User Not Authorized For Action	User Not Authorized For Action.

Syncpoint Participants GET

Retrieves a list of the Syncplicity users or user groups with whom the specified Syncpoint is shared. If the Syncpoint is already shared with a specified participant, the API will then behave as an update for the permissions instead. If an error occurred during the retrieval process, the API will return a non-200 HTTP status code.

Resource URL

`https://api.syncplicity.com/syncpoint/syncpoint_participants.svc/{SYNCPOINT_ID}/participants`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. The method's requestor should be a syncpoint owner, syncpoint sharing participant or the company admin

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request

Name	Value	Description
------	-------	-------------

Content-Type (required)	application/json	Content-Type of the request
-------------------------	------------------	-----------------------------

Sample JSON Response

```
[
  {
    "User": {
      "EmailAddress": "<User's Email>",
      "FirstName": "<User's First Name>",
      "LastName": "<User's Last Name>"
    },
    "Permission": 1,
    "Mapped": true
  },
  {
    "User": {
      "EmailAddress": "<User's Email>",
      "FirstName": "<User's First Name>",
      "LastName": "<User's Last Name>"
    },
    "Inviter": {
      "Id": "<User GUID>",
      "EmailAddress": "<User's Email>",
      "FirstName": "<User's First Name>",
      "LastName": "<User's Last Name>"
    },
    "Permission": 1,
    "Mapped": true
  },
  {
    "Group": {
      "Id": "<Group GUID>",
      "Name": "<Group Name>"
    },
    "Inviter": {
      "Id": "<User GUID>",
      "EmailAddress": "<User's Email>",
      "FirstName": "<User's First Name>",
      "LastName": "<User's Last Name>"
    },
    "Permission": 3,
    "Mapped": false
  }
]
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
User	List of users with whom the folder is shared	JSON Object	
User.EmailAddress	Email address of the syncpoint user participant	String (email)	
Group	List of groups with whom the folder is shared	JSON Object	
Group.Id	Unique ID associated with the participants in the group	GUID	
Permission	User or group permission for this syncpoint	Integer Enum	0 - None (Unshare, no permission to access the folder) 1 - ReadWrite (Read/write permission or the owner of the folder) 3 - ReadOnly (User can only download files from the Virtual Folder)
Mapped	Whether the syncpoint is mapped to a machine	Boolean	True or False

Response Error Details

HTTP Code	Error Code	Description
403	User Not Authorized For Action	

Syncpoint Participants POST

Shares a Syncpoint with one or more Syncplicity users or user groups. If the Syncpoint is already shared with the specified participant, the API will then behave as an update for the permissions instead. The API will return a non-200 HTTP status code.

Resource URL

```
https://api.syncplicity.com/syncpoint/syncpoint_participants.svc/{SYNCPOINT_ID}/participants?modifier={MODIFIER}
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. The method's requestor should be a syncpoint owner, syncpoint sharing participant or the company admin

URL Query Parameters

Name	Description
modifier	Set the value to "no_email" to suppress sharing notification emails

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample JSON Input

```
[
  {
    "User": {
      "EmailAddress": "<User's Email>"
    },
    "Permission": "3",
    "SharingInviteNote": "<Invitation note>"
  },
  {
    "Group": {
      "Id": "<Group Guid>"
    },
    "Permission": "1",
    "SharingInviteNote": "<Invitation note>"
  }
]
```

Sample JSON Input Definitions

Param Name	Required	Description	Data Type	Valid Values
User	Either User or Group should be defined, both cannot be missing	List of users to be added to the folder share	JSON Object	
User.EmailAddress	Yes if User is defined, otherwise no	Email address of the syncpoint user participant	String (email)	
Group	Either User or Group should be defined, both cannot be missing	List of groups to be added to the folder share	JSON Object	
Group.Id	Yes if Group is defined, otherwise no	Unique ID associated with the participants in the group	GUID	
Permission	Yes	User or group permission for this syncpoint	Integer Enum	0 - None (Unshare, no permission to access the folder) 1 - ReadWrite (Read/write permission or the owner of the folder) 3 - ReadOnly (User can only download files from the Virtual Folder)
SharingInviteNote	No	Message used for the invite email	String	

Sample JSON Response

```
[
  {
    "User": {
      "EmailAddress": "<User's Email>"
    },
    "Permission": "3",
    "SharingInviteNote": "<Invitation note>"
  },
  {
    "Group": {
      "Id": "<Group Guid>"
    },
    "Permission": "1",
    "SharingInviteNote": "<Invitation note>"
  }
]
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
User	List of users to be added to the folder share	JSON Object	
User.EmailAddress	Email address of the syncpoint user participant	String (email)	
Group	List of groups to be added to the folder share	JSON Object	
Group.Id	Unique ID associated with the group participant	GUID	
Permission	User or group permission for this syncpoint	Integer Enum	0 - None (Unshare, no permission to access the folder) 1 - ReadWrite (Read/write permission or the owner of the folder) 3 - ReadOnly (User can only download files from the Virtual Folder)
SharingInviteNote	Message used for the invite email	String	

Response Error Details

HTTP Code	Error Code	Description
403	User Not Authorized For Action	
403	Policy Prohibits Sharing	

Syncpoint Participants DELETE

Removes sharing participants in bulk. Un-shares a top-level Syncplicity folder with one or more users as specified in the request body.

Resource URL

`https://api.syncplicity.com/syncpoint/syncpoint_participants.svc/{SYNCPOINT_ID}/participants`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with target syncpoint	Integer	Positive integer value > 0. The method's requestor should be a syncpoint owner, syncpoint sharing participant or company admin

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample JSON Input

```
[
  {
    "User": {
      "EmailAddress": "<User's Email>"
    }
  },
  {
    "Group": {
      "Id": "<Group Guid>"
    }
  }
]
```

Sample JSON Input Definitions

Param Name	Required	Description	Data Type	Valid Values
User	Either User or Group should be defined, both cannot be missing	List of users to be added to the folder share	JSON Object	
User.EmailAddress	Yes if User is defined, otherwise no	Email address of the syncpoint user participant	String (email)	
Group	Either User or Group should be defined, both cannot be missing	List of groups to be added to the folder share	JSON Object	
Group.Id	Yes if Group is defined, otherwise no	Unique ID associated with the group participant	GUID	

Links POST

Creates and returns shared link to a file

Resource URL

`https://api.syncplicity.com/syncpoint/links.svc`

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample Input Definition

Field	Description	Data Type	Valid Values
Syncpoint ID	ID associated with the target syncpoint	Integer	Valid integer value > 0
VirtualPath	Virtual path to shared file	String	

Field	Description	Data Type	Valid Values
ShareLinkPolicy	Sharing policy	Integer	Unknown = 0 DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4
PasswordProtectPolicy	Password protected policy	Integer	Unknown = 0 Disabled = 1 Enabled = 2
Password	Link protection password	String	
Users	Array of users to share with	Array of users	
Groups	Array of groups to share with	Array of Groups	
Message	Message associated with the shared file	String	
OutlookShareLinkPolicy	Sharing policy for outlook	Integer	DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4

Sample JSON Input

```
[
  {
    "SyncpointId": <Syncpoint ID>,
    "VirtualPath": "<Virtual Path>",
    "SharedLinkPolicy": <Policy value>,
    "PasswordProtectPolicy": <Password protection policy>,
    "Password": "password"
    "Users": [ { "EmailAddress": <user email> } ],
    "Message": <text>,
    "OutlookSharePolicy": <Sharing policy for outlook>,
  },
  {
```

```

    "SyncpointId": <Syncpoint ID>,
    "VirtualPath": "<Virtual Path>",
    "SharedLinkPlicy": <Policy value>,
    "PasswordProtectPolicy": <Password protection policy>,
    "Password": "password"
    "Users": [ { "EmailAddress": <user email> } ],
    "Message": < text>,
    "OutlookSharePolicy": <Sharing policy for outlook>,
  },
  ...
]

```

Sample JSON Response definitions

Field	Description	Data Type	Valid Values
Syncpoint ID	ID associated with the target syncpoint	String	Valid integer value > 0
VirtualPath	Folder name	String	
ShareLinkPolicy	Sharing policy	Integer	Unknown = 0 DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4
PasswordProtectPolicy	Password protected policy	Integer	Unknown = 0 Disabled = 1 Enabled = 2
Users	Array of users to share with	Array of users	
DownloadUrl	Absolute URL for direct file download	String	Standard URL format
LinkUsage	Link usage statistics	Object	
LinkExpireInDays	Link expires after given amount of days	Integer	Valid integer value > 0

Field	Description	Data Type	Valid Values
IsExpired	True if link has been expired	Boolean	True or False
SharedDateUtc	UTC Date time when link was shared	DateTime	Valid UTC datetime
Token	Access token	String	Non-empty string
SharedType	Sharing type	Integer	AttachmentLink = 1 ShareLink = 2
LinkExpirationPolicy	Link expiration policy	Integer	Enabled = 1 Disabled = 2
LandingPageUrl	Landing page for shared file	String	Standard url format
File	Shared file information	Object	
Message	Message associated with the shared file	String	
OutlookShareLinkPolicy	Sharing policy for Outlook	Integer	DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4

Sample JSON Output

```
[
  {
    "SyncPointId": <Syncpoint ID>,
    "DownloadUrl": <Url for file download>,
    "SharedDateUtc": <Date time in UTC>,
    "Token": <access token>,
    "SharedType": <shared type>,
    "SharedLinkPlicy": <Policy value>,
    "LandingPageUrl": <Landing Page for shared file>
    "File": "File information"
    "PasswordProtectPolicy": <Password protection policy>,
    "Users": [],
    "Groups": [],
    "Message": <text>,
    "OutlookSharePolicy": <Sharing policy for outlook>
  },
  {
    "SyncPointId": <Syncpoint ID>,
    "DownloadUrl": <Url for file download>,
    "SharedDateUtc": <Date time in UTC>,
    "Token": <access token>,
    "SharedType": <shared type>,
    "SharedLinkPolicy": <Policy value>,
    "LandingPageUrl": <Landing Page for shared file>
    "File": "File information"
    "PasswordProtectPolicy": <Password protection policy>,
    "Users": []
    "Groups": []
    "Message": <text>,
    "OutlookSharePolicy": <Sharing policy for outlook>
  },
  ...
]
```

Response Error Details

HTTP Code	Error Code	Description
403	(Bad request) - spVirtualFolderGetByld did not return a result when one was expected	SyncPoint ID either invalid or not defined

Links GET

Returns a JSON array of shared links as well as information about link's protection and expiration time shared by given user.

Resource URL

```
https://api.syncplicity.com/syncpoint/links.svc/?user={USER_ID}&file_id={FILE_ID}
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{FILE_ID}	No	ID associated with the shared file	Long	Positive long value > 0
{USER_ID}	No	Unique user ID	GUID	If caller is authorized as Global Administrator it's possible to fetch shared links of other users by passing this parameter.

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token

Name	Value	Description
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample JSON Response definitions

Field	Description	Data Type	Valid Values
Syncpoint ID	ID associated with the target syncpoint	String	Valid integer value > 0
VirtualPath	Folder name	String	Non empty string
ShareLinkPolicy	Sharing policy	Integer	Unknown = 0 DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4
PasswordProtectPolicy	Password protected policy	Integer	Unknown = 0 Disabled = 1 Enabled = 2
Users	Array of users to share with	Array of users	
DownloadUrl	Absolute URL for direct file download	String	Standard url format
LinkUsage	Link usage statistics	Object	
LinkExpireInDays	Link expires after given amount of days	Integer	Valid integer value > 0
IsExpired	True if link has been expired	Boolean	True or False

Field	Description	Data Type	Valid Values
SharedDateUtc	UTC Date time when link was shared	DateTime	Valid UTC datetime
Token	Access token	String	Non-empty string
SharedType	Sharing type	Integer	AttachmentLink = 1 ShareLink = 2
LinkExpirationPolicy	Link expiration policy	Integer	Enabled = 1 Disabled = 2
LandingPageUrl	Landing page for shared file	String	Standard url format
File	Shared file information	Object	
Message	Message associated with the shared file	String	
OutlookShareLinkPolicy	Sharing policy for outlook	Integer	DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4

Sample JSON Output

```
[
  {
    "SyncPointId": <Syncpoint ID>,
    "DownloadUrl": <Url for file download>,
    "SharedDateUtc": <Date time in UTC>,
    "Token": <access token>,
    "SharedType": <shared type>,
    "SharedLinkPolicy": <Policy value>,
    "LandingPageUrl": <Landing Page for shared file>
    "File": "File information"
    "PasswordProtectPolicy": <Password protection policy>,
    "Users": [],
    "Groups": [],
    "Message": <text>,
    "OutlookSharePolicy": <Sharing policy for outlook>
  },
  {
    "SyncPointId": <Syncpoint ID>,
    "DownloadUrl": <Url for file download>,
    "SharedDateUtc": <Date time in UTC>,
    "Token": <access token>,
    "SharedType": <shared type>,
    "SharedLinkPolicy": <Policy value>,
    "LandingPageUrl": <Landing Page for shared file>
    "File": "File information"
    "PasswordProtectPolicy": <Password protection policy>,
    "Users": [],
    "Groups": [],
    "Message": <text>,
    "OutlookSharePolicy": <Sharing policy for outlook>
  },
  ...
]
```

Link PUT

Updates information associated with a shared link to a file.

Resource URL

`https://api.syncplicity.com/syncpoint/link.svc/{TOKEN}`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{TOKEN}	Yes	Shared link token	String	Valid token, non-empty string

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample Input Definition

Field	Description	Data Type	Valid Values
ShareLinkPolicy	Sharing policy	Integer	Unknown = 0 DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4
PasswordProtectPolicy	Password protected policy	Integer	Unknown = 0 Disabled = 1 Enabled = 2
Password	Link protection password	String	
Message	Message associated with the shared file	String	
OutlookShareLinkPolicy	Sharing policy for outlook	Integer	DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4

Sample JSON Input

```
{  
  "SharedLinkPolicy": <Policy value>,  
  "PasswordProtectPolicy": <Password protection policy>,  
  "Password": "password"  
  "Message": <text>,  
  "OutlookSharePolicy": <Sharing policy for outlook>  
}
```

Sample JSON Response definitions

Field	Description	Data Type	Valid Values
Syncpoint ID	ID associated with the target syncpoint	String	Valid integer value > 0

Field	Description	Data Type	Valid Values
VirtualPath	Folder name	String	
ShareLinkPolicy	Sharing policy	Integer	Unknown = 0 DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4
PasswordProtectPolicy	Password protected policy	Integer	Unknown = 0 Disabled = 1 Enabled = 2
DownloadUrl	Absolute URL for direct file download	String	
LinkUsage	Link usage statistics	Object	
LinkExpireInDays	Link expires after given amount of days	Integer	Valid integer value > 0
IsExpired	True if link has been expired	Boolean	True or False
SharedDateUtc	UTC Date time when link was shared	DateTime	Valid UTC datetime
Token	Access token	String	Non-empty string
SharedType	Sharing type	Integer	AttachmentLink = 1 ShareLink = 2
LinkExpirationPolicy	Link expiration policy	Integer	Enabled = 1 Disabled = 2
LandingPageUrl	Landing page for shared file	String	

Field	Description	Data Type	Valid Values
File	Shared file information	Object	
Message	Message associated with the shared file	String	
OutlookShareLinkPolicy	Sharing policy for outlook	Integer	DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4

Sample JSON Output

```
{
  "SyncPointId": <Syncpoint ID>,
  "DownloadUrl": <Url for file download>,
  "SharedDateUtc": <Date time in UTC>,
  "Token": <access token>,
  "SharedType": <shared type>,
  "SharedLinkPolicy": <Policy value>,
  "LandingPageUrl": <Landing Page for shared file>
  "File": "File information"
  "PasswordProtectPolicy": <Password protection policy>,
  "Password": "password"
  "Users": [],
  "Groups": [],
  "Message": <text>,
  "OutlookSharePolicy": <Sharing policy for outlook>
}
```

Response Error Details

HTTP Code	Error Code	Description
400	(Bad Request) Password policy Prohibits sharing	In case if PasswordProtectPolicy is enabled and password is empty
404	(Not Found) Token Not found	Access token is not valid

LinkGET

Get shared link information about a link's protection and expiration

Resource URL

`https://api.syncplicity.com/syncpoint/link.svc/{TOKEN}`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{TOKEN}	Yes	Shared link token	String	Valid token, non-empty string

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Sample JSON Response definitions

Field	Description	Data Type	Valid Values
Token	Shared file unique token	String	Non-empty string
SyncpointId	ID associated with the target syncpoint	Integer	Valid integer value > 0
VirtualPath	Path to shared file	String	
ShareLinkPolicy	Sharing policy	Integer	Unknown = 0 DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4
PasswordProtectPolicy	Password protected policy	Integer	Unknown = 0 Disabled = 1 Enabled = 2
Users	Array of users the link is shared with	Array of users	
DownloadUrl	Absolute URL for direct file download	String	Standard url format
LinkUsage	Link usage statistics	Object	
LinkExpireInDays	Link expires after given amount of days	Integer	Valid integer value > 0
IsExpired	True if link has been expired	Boolean	True or False
SharedDateUtc	UTC Date time when link was shared	DateTime	Valid UTC datetime
Token	Access token	String	Non-empty string

Field	Description	Data Type	Valid Values
SharedType	Sharing type	Integer	AttachmentLink = 1 ShareLink = 2
LinkExpirationPolicy	Link expiration policy	Integer	Enabled = 1 Disabled = 2
LandingPageUrl	Landing page for shared file	String	Standard URL format
File	Shared file information	Object	
Message	Message associated with the shared file	String	
OutlookShareLinkPolicy	Sharing policy for outlook	Integer	DisallowAll = 1 InternalOnly = 2 AllowAll = 3 IntendedOnly = 4

Sample JSON Output

```
{
  "SyncPointId": <Syncpoint ID>,
  "DownloadUrl": <Url for file download>,
  "SharedDateUtc": <Date time in UTC>,
  "Token": <access token>,
  "SharedType": <shared type>,
  "SharedLinkPolicy": <Policy value>,
  "LandingPageUrl": <Landing Page for shared file>
  "File": "File information"
  "PasswordProtectPolicy": <Password protection policy>,
  "Users": [],
  "Groups": [],
  "Message": <text>,
  "OutlookSharePolicy": <Sharing policy for outlook>
}
```

Response Error Details

HTTP Code	Error Code	Description
404	(Not Found) Token Not found	Access token is not valid

Link DELETE

Deletes shared link.

Resource URL

`https://api.syncplicity.com/syncpoint/link.svc/{TOKEN}`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{TOKEN}	Yes	Shared link token	String	Valid token, non-empty string

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	Content-Type of the request

Response Error Details

HTTP Code	Error Code	Description
404	(Not Found) Token Not found	Access token is not valid

Versions GET

Gets a list of versions for a specific file from a given syncpoint using the last known file version ID.

Resource URL

```
https://api.syncplicity.com/sync/versions.svc/{SYNCPOINT_ID}/file/{LAST_VERSION_ID}/versions
```

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. A private syncpoint will not be able to be referenced by this API
{LAST_VERSION_ID}	Yes	ID associated with the last known version	Long	Positive long value > 0

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user. If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.
AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token

Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Sample JSON Response

```
[
  {
    "SyncpointId": <Syncpoint ID>,
    "Id": <Version ID>,
    "UserName": <User name>,
    "DataSourceName": <Data source name>,
    "Action": <Action type>,
    "Length": <File Length>,
    "RevisionAge": "Revision age",
    "DateAddedUtc": <File upload date associated with latest version>
  },
  {
    "SyncpointId": <Syncpoint ID>,
    "Id": <Version ID>,
    "UserName": <User name>,
    "DataSourceName": <Data source name>,
    "Action": <Action type>,
    "Length": <File Length>,
    "RevisionAge": "Revision age",
    "DateAddedUtc": <File upload date associated with latest version>
  },
  ...
]
```

Sample JSON Response Definitions

Field	Description	Data Type	Valid Values
SyncpointId	ID associated with the folder syncpoint	Integer	Positive integer value > 0
Id	ID associated with the revision number	Long	Positive long value > 0
UserName	Name of the User who made changes	String	Non-empty string
DataSourceName	Data Source name (example: Web client)	String	Non-empty string

Field	Description	Data Type	Valid Values
Action	Action taken	Integer	0 - Added 1 - Edited 2 - Deleted 3 - Branched 4 - Renamed 5 - Confirmed Removed
Length	File size	Integer	Positive integer value > 0
RevisionAge	Date string (example: 10 days ago)	String	Non-empty string
DateAddedUtc	File upload date associated with latest version	DateTime	

Response Error Details

HTTP Code	Error Code	Description
400	Invalid SyncPoint ID	Invalid SyncPoint ID.
400	Invalid Version ID	Invalid Version ID.

Version DELETE

Deletes file version from given syncpoint by version ID.

Resource URL

`https://api.syncplicity.com/sync/version.svc/{SYNCPOINT_ID}/file/{FILE_ID}/version/{VERSION_ID}`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
{SYNCPOINT_ID}	Yes	ID associated with the target syncpoint	Integer	Positive integer value > 0. A private syncpoint will not be able to be referenced by this API
{FILE_ID}	Yes	ID associated with the file	Long	Positive long value > 0
{VERSION_ID}	Yes	ID associated with the file version ID	Integer	Positive integer value > 0

HTTP Header Parameters

Name	Value	Description
As-User (optional)	Valid user GUID	<p>A Global Administrator, or eDiscovery Administrator who has been assigned the corresponding privilege, can enter another user's GUID so that all Content API actions are performed on behalf of that user.</p> <p>If this header is not used or the Administrator has not been assigned the corresponding privilege, the API works on behalf of the currently authenticated Global or eDiscovery Administrator.</p>

AppKey (required)	<app_key>	Consumer Key of the app from Developer Portal
Authorization (required)	Bearer <token>	Authorization token
Accept (required)	application/json	Accept type of the request
Content-Type (required)	application/json	The Content-Type of the request

Response Error Details

HTTP Code	Error Code	Description
400	Invalid SyncPoint ID	Invalid SyncPoint ID.
400	Invalid Version ID	Invalid Version ID.

Chapter 3 API limits

This chapter describes API limits.

Name	Value	Affected services	Description
Entities per call	100	POST /provisioning/users.svc POST /provisioning/groups.svc POST /syncpoint/links.svc POST /syncpoint/participants.svc POST /syncpoint/syncpoint.svc	Limits total number of entities which could be proceeded through single request
Entities per call	20	POST /sync/files.svc POST /sync/folders.svc POST /sync/folder_folders.svc	Limits total number of entities which could be proceeded through single request
Groups per company	500 000	POST /provisioning/groups.svc	Max. number of groups per company supported by API
Members per groups	500 000	POST /provisioning/group_members.svc	Max. number of users per group supported by API
Links per User	10 000	POST /syncpoint/links.svc	Number of shared links created by single user
Participants per Syncpoint	1 000	POST /syncpoint/participants.svc	Max. participants number per single syncpoint
Syncpoints per User	200	POST /syncpoint/syncpoint.svc	Max. syncpoints number per single syncpoint
Syncpoints per company	10 000 000	POST /syncpoint/syncpoint.svc	Max. syncpoints number for all users in the same company
Files per User	1 000 000	POST /sync/files.svc	Max. number of active files available for user
Files per Syncpoint	500 000	POST /sync/files.svc	Max. number of active files per syncpoint

Files and Folders per directory	50 000	POST /sync/files.svc POST /sync/folders.svc POST /sync/folder_folders.svc	Max. number of active files and folders available per directory
Folders per Syncpoint	200 000	POST /sync/folders.svc POST /sync/folder_folders.svc	Max number of folders and subfolders available for single syncpoint

Chapter 4 ROL / GOL Awareness

This chapter describes the APIs necessary to write applications that are GOL/ROL aware.

OAuth Authentication

In general, the OAuth authentication mechanism is the same in the ROL and non-ROL use-cases. No code needs to change for the existing applications that were using the existing US API gateway `api.syncplicity.com`. However, in the case where a developer does want an application to invoke APIs on multiple ROLs, then it will be necessary to enhance the application in a few key areas.

- **AppKey/AppSecret:** The application being developed can use the same pair of **ApplicationKey/ApplicationSecret** for any ROL access through the API gateway. When you allocate a new **ApplicationKey/ApplicationSecret** in the developer portal, the **ApplicationKey/ApplicationSecret** are stored in all API gateways regardless of what ROL is being access, so as a developer you only need to track one pair of keys/secrets per application. NOTE: pre-existing **ApplicationKey/ApplicationSecret** that were defined pre-ROL/GOL awareness, will also have been updated in all the API gateways.

Bearer token: The bearer token returned from the initial OAuth login to the home ROL is valid on all ROLs, but the application **must** perform the initial OAuth authentication to only the home ROL, if another ROL is used for login/authentication the application will receive an authentication error and will not be able to proceed.

There is a distinction when the application must access the Personal Identifiable Information (PII) service of a foreign ROL, however; due to privacy reasons, a one-time bearer token will be allocated for access PII information on a foreign ROL. See more details in the PII service documentation below.

- **Refresh token:** refresh token can be used only with home ROL used for the initial authentication
- **Revoke token:** the revoke token call can only be submitted against the can be used only with issued ROL

ROIs Get

This is an unauthenticated API call to the GOL that resolves the list of ROIs that a given user has access to. The response is an array of ROIs with additional metadata informing which is the home ROI for the given user and what api gateway host name should be used to invoke API calls on that ROI.

Resource URL

`https://gol.syncplicity.com/api/v1/rois?uhid={user_email_hash}`

URL Parameters

Param Name	Required	Description	Data Type	Valid Values
uhid={user_email_hash}	Yes	If specified, retrieves the storage endpoint (StorageVault) definitions of the specified user; otherwise, the current user of the application is used.	String	User email hash is calculated using the user's email and encoding it using the HMAC MD5 algorithm with "6adebb9f-21f9-49d8-95bf-b7007a208cd4" salt

HTTP Header Parameters

Name	Value	Description
Accept (required)	application/json	Accept type of the request

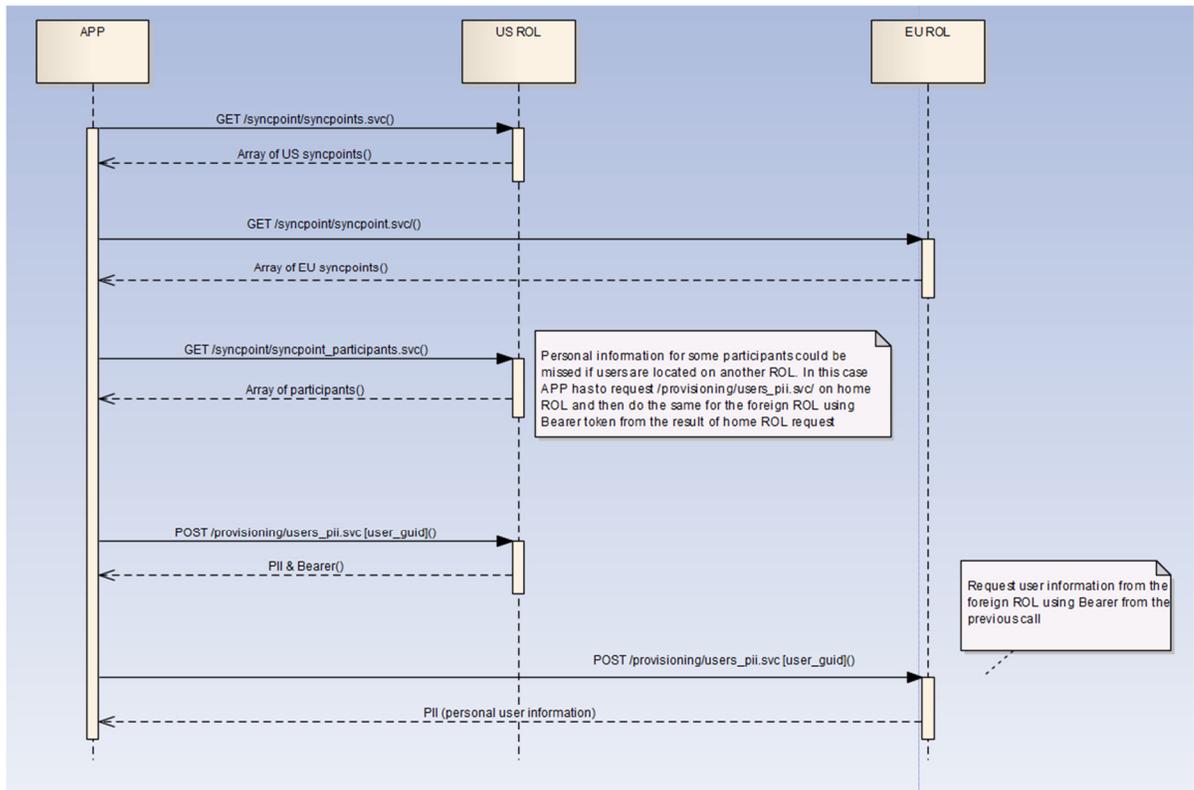
Personal Identifiable Information (PII)

To ensure clarity, an application can only look up PII information that they're authorized to see such as through a Syncpoint or file share between users of different ROLs. If the application's privilege's include accessing the participants of a particular Syncpoint or file-share then the application is authorized to look up the PII information of that user such as their name and email. No other personal information can be found for any unauthorized access.

For functional reasons, it may be necessary for the application to look up personal information that is stored in a different ROL. As an example, the Syncpoint-Participants API can return a list of users that participate in a given Syncpoint and one or more of these users may reside in a different home ROL from the application's. In this scenario, the foreign user's name and email will not be included in the response, instead a user id is the only metadata returned for these types of users. Programatically, an application can know that a particular user is from a foreign ROL by inspecting the `OriginalRolId` field of the various API responses, if the `OriginalRolId` value differs from the application's home ROL id, then the application knows which foreign ROL the user's PII resides.

If the application logic requires the PII information, the application must call the home ROL PII service with the user Ids to generate a one-time use bearer token that will be authorized to call the PII service of the foreign ROL to look up the user's name and email.

Below is a diagram that shows how the application may need to look up PII related data from a foreign ROL for the use-case mentioned above.



Example response from the syncpoint participants (See syncpoints participants documentation above for full API documentation).

- EmailAddress is empty string
- OriginalRolld differs from current ROL

Sample JSON Response

```
[
  {
    "User": {
      "EmailAddress": "someuser@dispostable.com",
      "FirstName": "someName",
      "LastName": "someLastName"
    },
    "Permission": 1,
    "Mapped": true,
    "IsOwner": true,
    "IsExplicit": true
  },
  {
    "User": {
      "Id": "bbe998e2-e759-495b-897e-6afaa76bb8c0",
      "EmailAddress": "",
      "FirstName": "",
      "LastName": "",
      "OriginalRoleId": 2
    },
    "Inviter": {
      "Id": "e3938cb5-e985-4fb1-be37-b4f2c8891477",
      "EmailAddress": "someuser@dispostable.com",
      "FirstName": "someName",
      "LastName": "someLastName"
    },
    "Permission": 1,
    "Mapped": false,
    "ResharingAllowed": false,
    "ParticipantResharingPolicy": 3,
    "IsExplicit": true
  }
]
```

Users PII GET

This API resolves the user GUIDs for the specified set of users into the PII related data for those users including First Name, Last Name and Email.

The API will provide results in the following use cases otherwise the call is considered unauthorized and will return an error:

- The application is sending the GUID of the currently authenticated user
- The application is authenticated as a Business Admin and the GUID represents a user which is a member of the company
- The application is authenticated a user that has shared a Syncpoint with a recipient that resides in the ROL being invoked and the GUID represents the Syncpoint participant residing in that ROL

If the PII service is invoked with a user GUID that is not a member of the ROL associated with the PII service call, but the application is authorized to view the PII information of the specified user, then the PII service will return a bearer token for the foreign ROL PII service to look up the PII information of the specified user in the foreign ROL PII service. This is a required step to look up PII related information for a user associated with a foreign ROL, the application must look up the bearer token in the application's home ROL PII service first, then make the remote ROL PII service call using the new bearer token.

Resource URL

`https://{ROL host name}/rol/users_pii.svc`

HTTP Header Parameters

Name	Value	Description
AppKey		Consumer Key of the app from Developer Portal. Note this must be specifically assigned to the ROL being invoked.
Authorization	Bearer <value>	Authorization token
Content-Type	application/json	The Content-Type of the request
Accept	application/json	Accept type of the request

Sample JSON Input

```
[
  "<user_guid_1>",
  "<user_guid_2>",
  ⋮
  "<user_guid_N>",
]
```

Sample JSON Input Definition

Accepts array of user GUIDs that will be processed and the service will return the PII data for each associated GUID. Invalid GUIDs will be ignored with no error response.

Field	Description	Data Type
Array	Array of user GUIDs	Array

Field	Description	Data Type
	grouped by the home ROL of the resulting users.	
RoUserPIIs.Users	Array of users with personal information inside	Array
RoUserPIIs.Users.Id	The guid of the returned user	GUID
RoUserPIIs.Users.EmailAddress	The email of the user if known on the current ROL	String
RoUserPIIs.Users.FirstName	The first name of the user if known on the current ROL	String
RoUserPIIs.Users.LastName	The first name of the user if known on the current ROL	String
RoUserPIIs.HomeRolId	The ID of the home ROL of the related users	Integer
RoUserPIIs.Bearer	One time use short-lived bearer token for invoking the PII service of the user's home ROL, if the associated users have a different home ROL from the current one.	String

Response Error Details

HTTP Code	Error Code	Description
403	Forbidden	
400	Bad request	User Ids Must Be Set, when there is no any user id in the list

This concludes the Content API Guide.