
KT ucloud storage service API

2017.1

문서 버전 및 이력

버전	일자	이력사항
	2010.01	
	2011.02	ACL 사용자 metadata 생성 추가
	2011.05	Large object 업로드 추가 S3 호환 API 추가
	2012.06	container static website 추가 container logging 추가
	2012.07	Reseller Service API 추가
	2012.11	사용자 metadata 삭제기능 추가
	2013.02	container 및 object name에 대한 제약사항 추가
	2013.03	container 및 object name에 대한 제약사항 변경 (제약조건 ':' 삭제)
	2013.04	S3 호환 API 분리
	2013.07	Jp-1에 대한 인증 URL 추가
	2013.10	Lite(KOR-HA) 인증 url 추가
	2103.11	TempURL, IPACL 추가
	2013.11	FormPost 추가
1.8.4	2013.12	Expiring Objects 추가
1.8.5	2015.04	Storage Large Object 추가
1.8.6	2015.11	11장 추가 (Cloudberry 공개권한 설정 추가)
1.8.7	2016.06	CORS 헤더 설정 추가 및 IP ACL 길이 확대
1.8.8	2017.1	최신 정보 업데이트

목 차

제 1 장 개요	5
1.1 KT ucloud storage service API 개요.....	5
1.2 ucloud storage service API 기능 개요.....	6
제 2 장 사용자 인증	8
2.1 사용자 인증.....	8
2.2 IP 인증제어.....	10
제 3 장 Storage account service	13
3.1 GET Storage account.....	13
3.2 HEAD Storage account.....	17
3.3 POST Storage account : 사용자 metadata.....	19
제 4 장 Storage container service	22
4.1 HEAD Storage container.....	22
4.2 GET Storage container.....	24
4.3 PUT Storage container.....	31
4.4 DELETE Storage container.....	33
4.5 POST Storage container : 사용자 metadata.....	35
4.6 POST Storage container : ACL.....	38
4.7 POST Storage container static website.....	40
4.8 POST Storage container logging.....	43
4.9 POST Storage container IP 접근제어.....	46
제 5 장 Storage object service	49
5.1 HEAD Storage object.....	49
5.2 GET Storage object.....	51
5.3 PUT Storage object.....	54
5.4 PUT Storage Large object.....	57
5.5 Chunked Transfer Encoding.....	60
5.6 Copy object.....	61
5.7 POST Storage object : 사용자 metadata.....	62
5.8 DELETE Storage object.....	65
제 6 장 S3 Compatible API	67
6.1 GET Service.....	67
6.2 GET Bucket.....	69
6.3 GET Object.....	72
6.4 HEAD Object.....	74

6.5 PUT Object	76
6.6 PUT Bucket	78
6.7 DELETE Bucket	80
6.8 DELETE Object.....	82
6.9 PUT Object Copy.....	84
제 7 장 Reseller service.	86
7.1 사용자 인증.....	86
7.2 GET account.....	88
7.3 GET user.....	90
7.4 GET users list.....	92
7.5 PUT user	94
7.6 DELETE user	97
제 8 장 TempURL Service	99
8.1 account 설정	99
8.2 시그니처 생성	99
8.3 TempURL 생성.....	100
8.4 TempURL 요청.....	100
제 9 장 FormPost Service	102
9.1 account 설정	102
9.2 시그니처 생성	102
9.3 FormPost 요청.....	103
제 10 장 Expiring Objects	105
10.1 Expiring Objects 설정.....	105
10.2 Expiring Objects 설정 해제	107
제 11 장 CORS(Cross Origin Resource Sharing)	108
11.1 CORS 메타데이터 설정.....	108
11.2 CORS 요청하기	108
11.3 CORS 실제 요청.....	109
제 12 장 Cloudberry 사용자를 위한 API 설정	110
12.1 POST storage container : ACL.....	110
제 13 장 API 사용 절차 및 웹 클라이언트 example	112
13.1 사용자 Token 및 서비스 URL 획득하기	112
13.2 일반적인 HTTP 요청.....	112
13.3 인증요청 자바 webclient sample code	113

제 1 장 개요

본 문서는 KT ucloud storage service API 규격을 제시한다.

본 API를 활용하여 클라우드 스토리지 응용서비스 개발을 위한 개발자 가이드 문서이다.

1.1 KT ucloud storage service API 개요

API는 웹 기반 ReSTful 방식으로 구현되었다. 모든 인증 및 container/object에 대한 요청은 표준 HTTP 호출을 통해서 이루어진다. ReST에 대한 자세한 정보는 Wikipedia를 참조하기 바란다.

다음은 ReST API's HTTP 요청에 대한 제약사항이다:

- HTTP 요청 헤더의 최대 개수 : 90
- HTTP 전체 헤더의 최대 길이: 4096 bytes
- HTTP 요청 line의 최대 길이: 8192 bytes
- HTTP 요청의 최대 길이: 5 gigabytes
- Container 이름의 최대 길이: 256 bytes
- Object 이름의 최대 길이: 1024 bytes

다음은 container 및 object name에 대한 제약사항이다

- 금지 특수 문자 : '\', '*', '(', '<', '>', '|'
- 금지 정규 표현식 : '/./', '/../', '/.\$', '/../.\$'

Container 및 Object의 이름은 먼저 적절하게 URL-encoded 되어야 하며, 또한 반드시 UTF-8 encoded 되어야 한다. 물론 위에서 언급했던 길이에 대한 제약 사항은 URL encoded 된 스트링에 대하여 수행해야 한다.

스토리지 서비스에 대한 각 HTTP 요청은 반드시 특정한 인증 token을 'X-Auth-Token'으로 정의된 HTTP 요청 헤더에 포함을 해야 한다. 모든 클라이언트는 유효한 Email주소와 API Key를 통해서 인증과정을 성공하면 Storage URL과 함께 인증 Token을 획득 할 수 있다.

1.2 ucloud storage service API 기능 개요

- 사용자 인증

기능	URI	설명
GET	<i>/storage/v1/auth</i>	사용자 인증요청
POST	<i>/account</i>	IP 인증제어

- Storage Accounts

기능	URI	설명
GET	<i>/account</i>	container 리스트 조회
HEAD		account metadata 조회
POST		account metadata 추가/변경/삭제

- Storage Containers

기능	URI	설명
HEAD	<i>/account/container</i>	container metadata 조회
GET		object 리스트 조회
PUT		container 생성
DELETE		container 삭제
POST		사용자 metadata 추가/변경/삭제 container ACL 설정 static website 구성 container 로깅 설정 container IP 접근제어

- Storage Objects

기능	URI	설명
HEAD	<i>/account/container/object</i>	object metadata 조회
GET		object 다운로드
PUT		object 업로드, object 복사
DELETE		object 삭제
POST		사용자 metadata 추가/삭제

- Reseller Service API

기능	URI	설명
GET	<i>/storage/v1/auth</i>	사용자 인증요청
	<i>/auth/v2/account</i>	account 정보조회
	<i>/auth/v2/account/userid</i>	사용자 정보조회
	<i>/auth/v2/account/.groups</i>	사용자 리스트 조회
PUT	<i>/auth/v2/account/userid</i>	사용자 생성
DELETE		사용자 삭제

제 2 장 사용자 인증

2.1 사용자 인증

- **API 설명**

사용자 인증을 요청한다.

사용자 인증은 포탈에 등록된 email주소와 포탈로부터 받은 API Key를 사용한다. 인증이 완료된 이후에는 response header를 통해서 storageUrl 및 Auth token이 전달된다. 인증 토큰의 유효시간은 최초 발행이후 86,400초(24시간)이며 응답헤더 X-Auth-Token-Expires에 잔여 유효시간이 표시된다. 또한 유효시간 내에는 유효시간이 단축된 동일한 인증 토큰이 발행되며, 새로운 인증 토큰 발행을 원할 경우 요청헤더 'X-Auth-New-Token'을 사용하면 된다(example참조). reseller service 인증은 7.1 장을 참고할 수 있다.

스토리지 상품에 대한 인증 URL은 아래와 같다.

Standard(KOR-Center)	https://api.ucloudbiz.olleh.com/storage/v1/auth https://ssproxy.ucloudbiz.olleh.com/auth/v1.0
Standard(JPN)	https://api.ucloudbiz.olleh.com/storage/v1/authjp https://ss.cloudk.whitecloud.jp/auth/v1.0

- **Request Syntax**

```
GET /storage/v1/auth HTTP/1.1
Host : 인증 서버
X-Storage-User : user_id(email 주소)
X-Storage-Pass : API key
```

'X-Storage-User'는 'X-Auth-User'로 대체 사용이 가능하며, 'X-Storage-Pass' 역시 'X-Auth-Key' 대체 사용이 가능하며 이들간의 혼용도 가능하다.

Request Parameters

사용자 인증 API에서는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes

X-Storage-User	user_id (email 주소)	String	Yes
X-Storage-Pass	포탈로부터 받은 API Key	String	Yes
X-Auth-Token-New-Token	신규 토큰 발급 요청	String	No

새로운 인증 토큰 발행을 원할 경우, 'X-Auth-New-Token' 헤더를 사용한다.

Request Body

사용자 인증 API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-Length	response body의 길이(bytes)
X-Storage-Url	스토리지 서비스 URL
X-Auth-Token	스토리지 서비스 사용 인증을 위한 토큰
X-Auth-Token-Expires	인증 토큰의 lifetime (단위: 초)

Response Body

json 스트링 타입으로 요청에 대한 결과를 제공한다.

제공하는 json key는 storage이며, storageUrl을 담고 있다.

자세한 예는 example을 참조할 것.

- **HTTP Status Code**

Description	HTTP Status code
인증 완료	200
인증 실패	401

- **Examples : 인증요청**

Sample Request

```
GET /storage/v1/auth HTTP/1.1
Host : api.ucloudbiz.olleh.com
X-Storage-User : test@test.com
X-Storage-Pass : API_KEY
```

Sample Response

```
HTTP/1.1 200 OK
Date: Mon, 21 Feb 2011 08:03:04 GMT
X-Storage-Url:      https://ssproxy.ucloudbiz.olleh.com/v1/AUTH_024067c0-f236-4d2a-80d0-736698fb6d36
X-Storage-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Auth-Token-Expires: 16154
Content-Length: 112
{"storage": {"default": "local", "local":
  "https://ssproxy.ucloudbiz.olleh.com/v1/AUTH_024067c0-f236-4d2a-80d0-736698fb6d36"}}
```

- **Examples : 신규 인증 토큰 발행 요청**

Sample Request

```
GET /storage/v1/auth HTTP/1.1
Host : api.ucloudbiz.olleh.com
X-Storage-User : test@test.com
X-Storage-Pass : API_KEY
X-Auth-New-Token:true
```

2.2 IP 인증제어

- **API 설명**

IP, 네트워크 기반으로 사용자 인증 요청을 제어한다.

사용자 인증 요청을 특정 IP 또는 네트워크로 제한 할 수 있다. IP 인증제어 기능은 최초 인증을 통해 정상적으로 storageUrl 및 Auth token을 받은 후 원하는 특정 IP 또는 네트워크를 등록하면 활성화 된다. 등록된 특정 IP 또는 네트워크에서만 사용자 인증 요청이 처리되고, 그 이외는 403을 응답한다. 사용자 실수로 잘못된 IP 또는 네트워크를 등록한 경우 인증 요청이 거절될 수 있다. 이 경우, ucloudbiz 포탈에 접속하여 정상 IP 및 네트워크로 수정하면 된다. IP 인증제어 기능을 불활성화 시키려면 등록된 IP 또는 네트워크 정보를 삭제하면 된다. (Example2 참조)

- **Request Syntax**

```
POST/<api version>/<account> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Account-Meta-Ipacl-Read : 특정 IP 또는 네트워크
X-Account-Meta-Ipacl-Write : 특정 IP 또는 네트워크
```

IP 인증제어는 'X-Account-Meta-Ipacl-Read' 및 'X-Account-Meta-Ipacl-Write'을 동일한 값으로 모두 설정을 해야 한다. IP 인증제어 disable 방법은 example2 참조 대상 IP/네트워크 지정 포맷은 '.n:IP 또는 .n:CIDR' 형식이다. 그 길이는 공백과 특수문자를 포함하여 512 byte로 제한한다.

(ex : X-Account-Meta-Ipacl-Read: .n:192.168.0.1,.n:192.168.0.2
X-Account-Meta-Ipacl-Write: .n:192.168.0.1,.n:192.168.0.2)

Request Parameters

IP 인증제어 API는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
Host	인증서버	String	Yes
X-Auth-Token	사용자 인증 Token	String	Yes
X-Account-Meta-Ipacl-Read	IP인증제어 요청 헤더	String	Yes
X-Account-Meta-Ipacl-Write			

Request Body

IP 인증제어 API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

- **HTTP Status Code**

Description	HTTP Status code
-------------	------------------

요청처리 완료	204
인증 실패(token error)	401
권한 없음	403

- **Examples1**

Sample Request

```
POST /v1/<account> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Account-Meta-Ipacl-Read: .n:192.168.0.1
X-Account-Meta-Ipacl-Write: .n:192.168.0.1
```

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
```

- **Examples2**

Sample Request

```
POST /v1/<account> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Remove-Account-Meta-Ipacl-Read: .n:192.168.0.1
X-Remove-Account-Meta-Ipacl-Write: .n:192.168.0.1
```

IP 인증제어 기능을 불활성화는 'X-Remove-Account-Meta-Ipacl-Read' 및 'X-Remove-Account-Meta-Ipacl-Write' 요청헤더를 이용한다. 요청이 정상처리되면 IP 인증제어를 위한 정보가 삭제되어 기능이 불활성화 된다. 요청 헤더의 값은 임

의의 값으로 처리해도 가능하다.

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
```

제 3 장 Storage account service

3.1 GET Storage account

- API 설명

container 리스트를 조회한다.

인증 API로부터 받은 X-Storage-Url을 통하여 현재 사용 중인 스토리지 container에 대한 리스트를 이름순으로 추출한다. 응답헤더에는 account 정보 및 사용자 metadata를 포함하고 있다.

- Request Syntax

```
GET /<api version>/<account>[?parameter=value] HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
```

Request Parameters

Parameter	Description	Required
Limit	container조회 결과에 최대 출력개수 지정 Type : 정수값	No
marker	주어진 string값보다 큰 name을 가진container 출력 Type : string	No
format	응답결과에 대한 출력 포맷 Type : string value : json or xml	No

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes

Request Body

스토리지 account API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
X-Account-Object-Count	대상 account의 전체 object의 개수
X-Account-Bytes-Used	대상 account의 전체 스토리지 사용량(bytes)
X-Account-Container-Count	대상 account의 총 container 개수
Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

Response Body

Examples 참조

- **HTTP Status Codes**

Description	HTTP Status code
처리완료(container 있음)	200
처리완료(container 없음)	204
인증 실패(token error)	401
권한 없음	403

- **Example1**

Sample Request

```
GET /v1/<account> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : AUTH_tk5f6d351c490b44b8b60b015e744a435a
```

Sample Response

```
HTTP/1.1 200 Ok
X-Account-Object-Count: 5
X-Account-Bytes-Used: 51055
X-Account-Container-Count: 4
Content-Length: 178
Content-Type: text/plain; charset=utf8
Date: Tue, 05 Apr 2011 01:28:39 GMT
```

```
documents
files
images
movies
```

- **Example2 – Serialized List Output(JSON)**

Sample Request

```
GET /v1/<account>?format=json HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
HTTP/1.1 200 Ok
X-Account-Object-Count: 3
X-Account-Bytes-Used: 95
X-Account-Container-Count: 2
Content-Length: 178
Content-Type: text/plain; charset=utf8
Date: Tue, 05 Apr 2011 01:28:39 GMT
```

```
[
    {"name":"test_container_1", "count":2, "bytes":78},
    {"name":"test_container_2", "count":1, "bytes":17}
]
```

- **Example3 – Serialized List Output(XML)**

Sample Request

```
GET /v1/<account>?format=xml HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 200 Ok
X-Account-Object-Count: 2
X-Account-Bytes-Used: 78
X-Account-Container-Count: 1
Content-Length: 178
Content-Type: text/plain; charset=utf8
Date: Tue, 05 Apr 2011 01:28:39 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<account name="MichaelBarton">
  <container>
    <name>test_container_1</name>
    <count>2</count>
    <bytes>78</bytes>
  </container>
</account>
```

- **Example4 – Large Container Lists**

시스템은 매 요청마다 최대 1,000개의 container를 return한다.
따라서 limit 및 marker 파라미터를 이용하여 원하는 결과를 받을 수 있다.
만일, 전체 container를 아래와 같이 5개가 있다고 가정하자.

```
apples
bananas
kiwis
oranges
pears
```


Sample Request

```
GET /v1/<account>?limit=2&marker=bananas HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

위 요청에서 최대 결과값 반환을 2개로 제한하고 있으며, 또한 marker를 이용해서 bananas 보다 큰 값(문자열 순서에 따라)을 가진 container 이름을 반환한다.

```
kiwis
oranges
```

3.2 HEAD Storage account

account metadata를 조회한다.

해당 account의 현재 저장된 object 개수, container개수 및 전체 저장된 용량(bytes) 및 사용자 metadata 정보를 가져온다.

- **Request Syntax**

```
HEAD /<api version>/<account> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
```

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes

Request Body

HEAD스토키지 account API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
X-Account-Object-Count	대상 account의 전체 object의 개수
X-Account-Bytes-Used	대상 account의 전체 스토리지 사용량(bytes)
X-Account-Container-Count	대상 account의 총 container 개수

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
처리완료	204
인증 실패(token error)	401
권한 없음	403

- **Example**

Sample Request

```
HEAD /v1/<account> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 204 No Content
X-Account-Object-Count: 2
X-Account-Bytes-Used: 78
X-Account-Container-Count: 1
Content-Length: 178
Content-Type: text/plain; charset=utf8
Date: Tue, 05 Apr 2011 01:28:39 GMT
```

3.3 POST Storage account : 사용자 metadata

- API 설명

account에 metadata를 추가 또는 수정한다.

'X-Account-Meta-' 접두어를 갖는 key/value 형태의 account 사용자 metadata를 추가하거나 덮어쓰기를 수행한다. 즉, 이미 생성한 account 사용자 metadata에 대해서는 수정 작업을 한다. X-Account-Container-Count, X-Account-Total-Bytes-Used 와 같은 header는 수정불가하다. 기 추가한 account metadata의 삭제는 'X-Remove-Account-Meta-' 접두어를 사용하여 삭제하며 구체적인 사용예는 Example2를 참조한다.

- Request Syntax

```
POST/<api version>/<account> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Account-Meta-key: value
```

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Account-Meta-*	추가/변경하고자 하는 사용자 metadata	String	Yes

(*) 임의의 String

Request Body

POST 스토리지 account API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식
Content-Length	response body의 길이

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	204
인증 실패(token error)	401
권한 없음	403

- **Example1**

Sample Request

```
POST /v1/<account> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Account-Meta-Fruit: Test1
X-Account-Meta-Veggie: Test2
```

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

추가된 account 사용자 metadata는 GET 또는 HEAD로 조회가 가능하며, 기존에 추가한 사용자 metadata에 대해서 동일한 POST operation을 통해서 값을 수정할 수 있다.

- **Example2**

Example1 에서 추가한 account metadata 를 삭제하는 예이다.

```
POST /v1/<account> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Remove-Account-Meta-Fruit: Test1
X-Remove-Account-Meta-Veggie: Test2
```

기 생성한 사용자 metadata의 key에 Remove를 추가하여 요청하면 된다. 그러나 삭제 요청 key에 대한 value는 반드시 이전 value로 해야 할 필요는 없고, 임의의 string으로 정의해도 된다.

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 01 Nov 2012 09:07:11 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

제 4 장 Storage container service

4.1 HEAD Storage container

- API 설명

container metadata를 조회한다.

특정 container에 대해서 저장되어 있는 object 개수와 전체 저장된 object의 저장 용량(bytes) 및 기 추가한 container 사용자 metadata 를 조회한다.

- Request Syntax

```
HEAD /<api version>/<account>/<container> HTTP/1.1
```

```
Host : 스토리지 서비스 서버
```

```
X-Auth-Token : 인증 토큰
```

Required Request Headers

Header Name	Description	Type	Required
-------------	-------------	------	----------

X-Auth-Token	사용자 인증 Token	String	Yes
--------------	--------------	--------	-----

Request Body

HEAD스토리지 container API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식
X-Container-Object-Count	container에 저장된 object 개수
X-Container-Bytes-Used	전체 저장된 object의 용량(bytes)

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
처리완료	204
인증 실패(token error)	401
권한 없음	403
container 없음	404

- **Example**

Sample Request

```
HEAD /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 204 No Content
Date: Mon, 12 Nov 2007 15:32:21 GMT
Content-type: text/html
X-Container-Object-Count: 3
X-Container-Bytes-Used : 32456
X-Container-Meta-key1: value1
X-Container-Meta-key2: value2
```

4.2 GET Storage container

- API 설명

container에 저장된 object의 리스트를 조회한다.

특정 container에 저장된 object의 리스트 및 해당 container의 정보 및 추가한 container 사용자 metadata를 가져온다.

- Request Syntax

```
GET /<api version>/<account>/<container>[?parameter=value] HTTP/1.1
```

```
Host : 스토리지 서비스 서버
```

```
X-Auth-Token : 인증 토큰
```

Request Parameters

Parameter	Description	Required
limit	container조회 결과에 최대 출력개수 지정	No

	Type : 정수값	
marker	주어진 string값보다 큰 name을 가진container 출력 Type : string	No
prefix	주어진 string값으로 시작하는 object를 출력 Type : string	No
format	응답결과에 대한 출력 포맷 Type : string value : json or xml	No
path	pseudo path example 내용 참조	No

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes

Request Body

GET 스토리지 container API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Server	서비스 서버
X-Container-Object-Count	container에 저장된 object 개수
X-Container-Bytes-Used	전체 저장된 object의 용량(bytes)
Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

Response Body

Examples 참조

- **HTTP Status Codes**

Description	HTTP Status code
처리완료(object 있음)	200
처리완료(object/container 없음)	204

인증 실패(token error)	401
권한 없음	403
account 오류	404

- **Example1**

Sample Request

```
GET /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 200 Ok
X-Container-Object-Count: 7
X-Container-Bytes-Used : 312456
X-Container-Meta-key1: value1
X-Container-Meta-key2: value2
Content-Length: 32
Content-Type: text/plain; charset=UTF-8
Date: Mon, 12 Nov 2007 15:32:21 GMT
```

```
kate_beckinsale.jpg
How To Win Friends And Influence People.pdf
moms_birthday.jpg
poodle_strut.mov
Disturbed - Down With The Sickness.mp3
army_of_darkness.avi
the mad.avi
```

- **Example2 – Serialized List Output(JSON)**

Sample Request

```
GET /v1/<account>/<container>?format=json HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
Content-Length: 0
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 200 Ok
X-Container-Object-Count: 2
X-Container-Bytes-Used : 78
Content-Length: 32
Content-Type: text/plain; charset=UTF-8
Date: Mon, 12 Nov 2007 15:32:21 GMT
```

```
[
  {"name":"test_obj_1",
    "hash":"4281c348eaf83e70ddce0e07221c3d28",
    "bytes":14,
    "content_type":"applicationW/octet-stream",
    "last_modified":"2009-02-03T05:26:32.612278"},
  {"name":"test_obj_2",
    "hash":"b039efe731ad111bc1b0ef221c3849d0",
    "bytes":64,
    "content_type":"applicationW/octet-stream",
    "last_modified":"2009-02-03T05:26:32.612278"},
]
```

- **Example3 – Serialized List Output(XML)**

Sample Request

```
GET /v1/<account>/<container>?format=xml HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 200 Ok
X-Container-Object-Count: 2
X-Container-Bytes-Used : 78
Content-Length: 32
Content-Type: application/xml; charset=UTF-8
Date: Mon, 12 Nov 2007 15:32:21 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<container name="test_container_1">
  <object>
    <name>test_object_1</name>
    <hash>4281c348eaf83e70ddce0e07221c3d28</hash>
    <bytes>14</bytes>
```

- **Example4 – Large Object Lists**

시스템은 매 요청마다 최대 1,000개의 object를 return한다.

따라서 limit 및 marker 파라미터를 이용하여 원하는 결과를 받을 수 있다.

만일, 전체 object를 아래와 같이 5개가 있다고 가정하자.

```
apples  
bananas  
kiwis  
oranges  
pears
```

Sample Request

```
GET /v1/<account>/<container>?limit=2&marker=oranges HTTP/1.1  
Host : ssproxy.ucloudbiz.olleh.com  
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

위 요청에서 최대 결과값 반환을 2개로 제한하고 있으며, 또한 marker를 이용해서 oranges 보다 큰 값(문자열 순서에 따라)을 가진 object 이름을 반환한다.

```
HTTP/1.1 200 Ok
X-Container-Object-Count: 5
X-Container-Bytes-Used : 3456
Content-Length: 32
Content-Type: application/xml; charset=UTF-8
Date: Mon, 12 Nov 2007 15:32:21 GMT
```

```
pears
```

- **Example5 – Pseudo hierarchical folders/directories**

KT Ucloud Storage 저장구조는 사용자들에게 친숙한 계층구조를 갖는 파일시스템은 아니다. 그러나, 아래와 같은 2 가지 방법으로 유사하게 계층구조를 갖는 저장공간처럼 유사하게 서비스 제공이 가능하다.

- (1) object 이름에 반드시 path element separator(/)를 가져야 한다.
- (2) '*directory marker*' object 를 생성해야 한다.

Sample Objects

아래와 같이, 실제 object들이 저장되어 있다고 가정하자.

마치, 계층구조를 갖는 것처럼 object의 이름에 '/' 포함되어 있으나 본질적으로는 순수하게 object의 name이다.

예를 위해 모두 backups container에 저장되어 있다고 가정하자

```
photos/animals/dogs/poodle.jpg
photos/animals/dogs/terrier.jpg
photos/animals/cats/persian.jpg
photos/animals/cats/siamese.jpg
photos/plants/fern.jpg
photos/plants/rose.jpg
photos/me.jpg
```

Sample Directory Marker Objects

이제 계층구조와 같이 유사하게 처리하기 위하여, directory marker object를 생성해야 한다. 즉, 시스템에서 디렉토리로 처리하기 위한 표시를 만들기 위하여 별도의 object를 생성하는 것이다.

(object의 크기는 0, Content-type은 application/directory)

```
photos/animals/dogs
photos/animals/cats
photos/animals
photos/plants
photos
```

Sample GET Requests/Responses

'path' query parameter를 이용하여 request 및 response에 대한 예이다.

일부 header 내용은 생략하였다.

```
GET /v1/<account>/backups?path=photos HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
photos/animals
photos/plants
photos/me.jpg
```

```
GET /v1/<account>/backups?path=photos/animals HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
photos/animals/dogs
photos/animals/cats
```

4.3 PUT Storage container

- **API 설명**

container를 생성한다.

container를 생성하거나, container 사용자 metadata를 추가하거나 수정한다.
사용자 metadata 추가 및 수정은 POST API 참조.

- **Request Syntax**

PUT /<api version>/<account>/<container> HTTP/1.1

Host : 스토리지 서비스 서버

X-Auth-Token : 인증 토큰

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes

Request Body

PUT스토리지 container API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
container 생성	201
생성 요청한container 이미 존재	202
인증 실패(token error)	401
권한 없음	403

- **Example**

Sample Request

```
PUT /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : AUTH_tk5f6d351c490b44b8b60b015e744a435a
```

Sample Response

```
HTTP/1.1 201 Created
Content-type: text/plain; charset=UTF-8
```


4.4 DELETE Storage container

- API 설명

container를 삭제한다.

container가 삭제 가능하기 위해서는 저장된 object가 없어야 한다. 즉, 저장되어 있는 object를 모두 삭제하고 마지막으로 container를 삭제한다.

- Request Syntax

```
DELETE /<api version>/<account>/<container> HTTP/1.1
```

Host : 스토리지 서비스 서버

X-Auth-Token : 인증 토큰

Required Request Headers

Header Name	Description	Type	Required
-------------	-------------	------	----------

X-Auth-Token	사용자 인증 Token	String	Yes
--------------	--------------	--------	-----

Request Body

DELETE 스토리지 container API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
container 삭제 완료	204
인증 실패(token error)	401
권한 없음	403
삭제 요청한container 없음	404
삭제 요청한container 저장된 object 있음 삭제 불가	409

- **Example**

Sample Request

<pre>DELETE /v1/<account>/<container> HTTP/1.1 Host : ssproxy.ucloudbiz.olleh.com X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb</pre>

Sample Response

<pre>HTTP/1.1 204 No Content Date: Thu, 07 Jun 2007 18:57:07 GMT Content-Length: 0 Content-Type: text/plain; charset=UTF-8</pre>
--

4.5 POST Storage container : 사용자 metadata

- API 설명

container metadata를 추가 또는 수정한다.

'X-Container-Meta-' 접두어를 갖는 key/value 형태의 container metadata를 추가하거나 덮어쓰기를 수행한다. 즉, 이미 추가한 container metadata에 대해서는 수정 작업을 수행한다. X-Container-Object-Count, X-Container-Total-Bytes-Used 와 같은 header는 수정불가하다. 기 추가한 container metadata의 삭제는 'X-Remove-Container-Meta-' 접두어를 사용하여 삭제하며 구체적인 사용예는 Example2를 참조한다.

- Request Syntax

```
POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Container-Meta-key: value
```

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Container-Meta-*	추가/변경하고자 하는 Metadata	String	Yes

(*) 임의의 String

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식
Content-Length	response body의 길이

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	204
인증 실패(token error)	401
권한 없음	403

- **Example1**

Sample Request

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Container-Meta-Fruit: Test1
X-Container-Meta-Veggie: Test2
```

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

- **Example2**

Example1 에서 추가한 container metadata 를 삭제하는 예이다.

Sample Request

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Remove-Container-Meta-Fruit: Test1
X-Remove-Container-Meta-Veggie: Test2
```

기존 metadata 생성 header key에 Remove를 추가하여 요청하면 된다. 그러나 삭제요청 key에 대한 value는 반드시 이전 value로 해야 할 필요는 없고, 임의의 string으로 정의해도 된다.

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 01 Nov 2012 09:07:11 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

4.6 POST Storage container : ACL

- API 설명

container의 ACL을 설정한다.

KT Ucloud storage service는 container별로 접근제어에 대한 설정이 가능하다. http post를 사용하여 사용자 metadata를 정의하는 방식과 동일하나, 특정 헤더 key를 사용해야한다. ACL을 위해서는 두 개의 header key를 사용하는데 읽기에 대한 제어는 'x-container-read' 키를 사용하고, 쓰기에 대한 제어는 'x-container-write' 키를 사용해야만 한다. header key에 대한 header value의 형식은 [item[,item...]]와 같다. 각 item은 권한을 주고자 하는 '계정의 이름' 이거나 HTTP Referrer header에 근거한 제어를 하기 위해 'referrer 지정'의 두 가지 내용을 갖고, 이러한 작업을 통해서 ACL을 추가하거나 변경이 가능하다.

자세한 내용은 아래 example을 참조

	example	description
referrer 형식	.r.*	임의의 host 접근 허용
	.r.www.example.com	특정 host 허용
	.r*.example.com	특정 domain 허용

	.r:-example.com	특정 domain 거절
	r:-thief.example.com	특정 host거절
그룹 지정	bobs_account	특정 account 허용
	sues_account:sue (계정이름:유저1이름)	특정 사용자 허용

- **Request Syntax**

POST/<api version>/<account>/<container> HTTP/1.1

Host : 스토리지 서비스 서버

X-Auth-Token : 인증 토큰

x-container-read (x-container-write) : [item[,item...]]

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
x-container-read/write	ACL을 설정하기 위한 http header key	String	Yes
X-Remove-Container-Read / write	ACL 권한 회수를 위한	String	Yes

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식
Content-Length	response body의 길이

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
-------------	------------------

요청처리 완료	204
인증 실패(token error)	401
권한 없음	403

- **Example**

Sample Request

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : 'X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
x-container-read: .r:.example.com,.r:-thief.example.com
x-container-write: bobs_account,sues_account:sue
```

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

4.7 POST Storage container static website

- **API 설명**

container를 **static website**로 구성한다.

KT ucloud storage service는 container를 이용하여 별도의 웹서버 구축없이 static web 서비스가 가능하다. 우선적으로 container를 static website로 사용하기 위해서는 먼저 container가 읽기권한으로 공개되어야 한다(POST Storage container : ACL api 참조). 본 api는 특정 container를 static website로 설정하는 기능을 제공한다. Static Web 서비스에 대한 이용가이드는 다음 링크를 참조한다.

https://ucloudbiz.olleh.com/manual/ucloud_storage_Static_Web_service_user_guide.pdf

- **Static website index 페이지 설정**

Request Syntax

```
POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Container-Meta-Web-Index: 해당 container 내 오브젝트 파일 (ex. index.html)
```


Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Container-Meta-Web-Index	static website에 대한 index 파일 지정(임의선택 가능)	String	Yes

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- **Static website error 페이지 설정**

우선 error 페이지는 2 가지를 지원한다. 401(not authorized), 404(not found) 이때 suffix 를 (error.html)이라고 지정하면, 만약 401 에러시 401error.html 파일을 호출하고 404 에러시 404error.html 파일을 호출하게 된다.

Request Syntax

```
POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Container-Meta-Web-Error: 해당 container 내 에러 파일 suffix (ex. error.html)
```

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Container-Meta-Web-Error	static website의 에러 페이지에 대한 suffix지정(임의선택 가능)	String	Yes

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- **HTML file listings 설정**

index.html 페이지를 설정하지 않았을 때 해당 static website container내 오브젝트 (ex. html 파일리스트) 리스트를 보여주고 선택할 수 있게 설정 가능하며 style sheet 지정도 가능하다.

Request Syntax

```
POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Container-Meta-Web-Listings: TRUE_VALUE
X-Container-Meta-Web-Listings-Css: 해당 container 내 style sheet 파일
                                (ex. listing.css)
```

TRUE_VALUE = (true, 1, yes, on, t, y) 중에서 임의의 1 개 지정 가능

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Container-Meta-Web-Listings	static website의 HTML 파일 리스트 출력 설정	enum	Yes
X-Container-Meta-Web-Listings-Css	style sheet 지정	String	Yes

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- HTTP Status Codes

Description	HTTP Status code
요청처리 완료	204
인증 실패(token error)	401
권한 없음	403

- Example

Static website index페이지 설정

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Container-Meta-Web-Index: index.html
```

Static website error suffix 설정

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Container-Meta-Web-Error: error.html
```

HTML file listings 설정

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Container-Meta-Web-Listings: true
X-Container-Meta-Web-Listings-Css: listing.css
```

- Container static website 구성을 위한 metadata 삭제 및 수정은 POST Storage container : 사용자 metadata API를 참조하면 된다.

4.8 POST Storage container logging

- API 설명

container 로깅을 구성한다.

KT Ucloud storage service는 container별로 접근로그를 제공한다. 본 api는 container에 대한 접근정보를 로그파일로 저장을 원할 경우 필요한 설정을 위한 기능을 제공한다. 로그저장 기능을 활성화시킬 경우, 해당 계정내에 .ACCESS_LOGS 라는 container가 자동으로 생성되며 해당 container에 로그 파일이 저장되며 gzip 형식의 압축파일로 저장된다. 단, 로그저장 기능을 설정하지 않은 container는 로그가 생성되지 않는다.

로그 파일명은 container/YYYY/MM/DD/HH/uuid.log.gz

- container: '로그저장' 기능을 신청한 container명
- YYYY: year, MM: month, DD: day, HH: hour로그파일을 필터링한 시간 정보
- uuid: 랜덤하게 만들어지는 수(16진수 32자)
- log: 이 파일이 로그파일임을 나타내는 확장자

- gz: 생성된 로그 파일은 gzip으로 압축하여 저장됨을 나타냄

로그저장 서비스 이용에 대한 자세한 가이드는 아래 링크를 참조한다.

https://ucloudbiz.olleh.com/manual/ucloud_storage_log_save_service_user_guide.pdf

- **로그저장 기능 활성화**

Request syntax

```
POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Container-Meta-Access-Log-Delivery: TRUE_VALUE
```

TRUE_VALUE = (true, 1, yes, on, t, y) 중에서 임의의 1 개 지정 가능

Required Request Headers

Header Name	Description	Type	default
X-Auth-Token	사용자 인증 Token	String	없음
x-container-meta-access-log-delivery	로그저장 기능 활성화/비활성화 지정	enum	없음

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- **로그저장 기능 비활성화**

Request syntax

```
POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
x-container-meta-access-log-delivery: TRUE_VALUE 가 아닌 값
```

Required Request Headers

Header Name	Description	Type	default
X-Auth-Token	사용자 인증 Token	String	없음
x-container-meta-	로그저장 기능	String	없음

access-log-delivery	활성화/비활성화 지정		
---------------------	-------------	--	--

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	204
인증 실패(token error)	401
권한 없음	403

- **Example**

로그저장 활성화

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : 'X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
x-container-meta-access-log-delivery: true
```

로그저장 비활성화

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : 'X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Container-Meta-Access-Log-Delivery: false
```

container logging 구성을 위한 metadata 삭제 및 수정은 POST Storage container : 사용자 metadata API를 참조하면 된다.

4.9 POST Storage container IP 접근제어

- **API 설명**

IP, 네트워크 기반으로 Container에 대한 요청을 제어한다.

컨테이너 정보 조회, 다운로드 또는 업로드를 특정 IP 또는 네트워크로 제한 할 수 있다. Container IP 접근제어 기능은 최초 인증을 통해 정상적으로 storageUrl 및 Auth token을 받은 후 원하는 특정 IP 또는 네트워크를 등록하면 기능이 활성화 된다. 등록된 특정 IP 또는 네트워크에서만 파일 리스트 조회, 다운로드 또는 업로드 요청이 처리되고, 그 이외는 403을 응답한다. 사용자 실수로 잘못된 IP 또는 네트워크를 등록한 경우 Container에 대한 요청이 거절될 수 있다. 이 경우, ucloudbiz 포탈에 접속하여 정상 IP 및 네트워크로 수정하면 된다. Container IP 접근제어 기능을 불활성화 시키려면 등록된 IP 또는 네트워크 정보를 삭제하면 된다. (Example2 참조)

- **Request Syntax**

```

POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
X-Container-Meta-Ipacl-Read : 특정 IP 또는 네트워크
X-Container-Meta-Ipacl-Write : 특정 IP 또는 네트워크

```

'X-Container-Meta-Ipacl-Read'는 Container 정보 조회 또는 다운로드 제어,
 'X-Container-Meta-Ipacl-Write'는 다운로드 또는 사용자 metadata 추가 요청을
 제어한다. 사용 목적에 따라 각각 또는 동시에 설정하여 사용이 가능하다.
 대상 IP/네트워크 지정 포맷은 '.n:IP 또는 .n:CIDR' 형식이다.
 (ex : X-Container-Meta-Ipacl-Read: .n:192.168.0.1,.n:192.168.0.2
 X-Container-Meta-Ipacl-Write: .n:192.168.0.1,.n:192.168.0.2)

Request Parameters

Container IP 접근제어 API는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes
X-Auth-Token	사용자 인증 Token	String	Yes
X-Container-Meta-Ipacl-Read	IP 접근제어 요청 헤더	String	No
X-Container-Meta-Ipacl-Write			

Request Body

Container IP 접근제어 AP는 request body를 사용하지 않는다.

● **Response**

Response Headers

Header Name	Description
Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

● **HTTP Status Code**

Description	HTTP Status code
요청처리 완료	204

인증 실패(token error)	401
권한 없음	403

- **Examples1**

Sample Request

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Container-Meta-Ipacl-Read: .n:192.168.0.1
X-Container-Meta-Ipacl-Write: .n:192.168.0.3
```

IP 192.168.0.1 에는 해당 컨테이너 정보조회 및 다운로드 허용

IP 192.168.0.3 에는 해당 컨테이너에 업로드 허용

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
```

- **Examples2**

Sample Request

```
POST /v1/<account> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Remove-Container-Meta-Ipacl-Read: .n:192.168.0.1
X-Remove-Container-Meta-Ipacl-Write: .n:192.168.0.3
```

IP 인증제어 기능을 불활성화는 'X-Remove-Account-Meta-Ipacl-Read' 및

'X-Remove-Account-Meta-Ipacl-Write' 요청헤더를 이용한다. 요청이 정상처리되면

IP 인증제어를 위한 정보가 삭제되어 기능이 불활성화 된다. 삭제를 위해서는 요청 헤더의 값은 임의의 string이 가능하다.

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
```


제 5 장 Storage object service

5.1 HEAD Storage object

- API 설명

object metadata를 조회한다.

object에 대한 사용자 metadata 및 http response header 정보를 추출한다.
(object 사용자 metadata 추가에 대해서는 POST Storage object API 참조)

- Request Syntax

```
HEAD /<api version>/<account>/<container>/<object> HTTP/1.1  
Host : 스토리지 서비스 서버  
X-Auth-Token : 인증 토큰
```

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes

Request Body

HEAD 스토리지 object API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Last-Modified	마지막 수정일
Etag	MD5 checksum
Content-Type	response body의 형식
Content-Length	response body의 길이
X-Object-Meta-*	사용자 metadata

(*) 임의의 String

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	200
인증 실패(token error)	401
권한 없음	403
요청한 object 존재하지 않음	404

- **Example**

Sample Request

```
HEAD /v1/<account>/<container>/<object> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
sX-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 07 Jun 2007 18:57:07 GMT
Etag: 8a964ee2a5e88be344f36c22562a6486
Content-Length: 5120
Content-Type: binary/octet-stream
X-Object-Meta-Meat: Bacon
X-Object-Meta-Fruit: Bacon
```

5.2 GET Storage object

- API 설명

object를 다운로드한다.

object data 및 사용자 metadata를 가져온다.

- Request Syntax

GET /<api version>/<account>/<container>/<object> HTTP/1.1

Host : 스토리지 서비스 서버

X-Auth-Token : 인증 토큰

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	없음

Optional Request Headers

Header Name	Description	Type	default
Range	부분 data 지정 (자세한 내용 아래참조)	integer	없음

If-Match	Etag값 비교하여 수행결과 반환 (자세한 내용 아래 참조)		
If-None-Match	ex) If-Match: 676e05583d86d74622522d9817f995d1		
If-Modified-Since	헤더값에 지정된 날짜보다 나중자원 전달 ex)If-Modified-Since: Tue, 15 Nov 1994 12:45:26 GMT		
If-Unmodified-Since	헤더값에 지정된 날짜로부터 수정이 없는경우 Method를 수행 ex) If-Unmodified-Since: Sat, 29 Oct 1994 19:43:31 GMT		

Range 헤더는 RFC2616의 전체 spec을 지원하지는 않는다. OFFSET-LENGTH 형태의 값을 지원하다. 두 값은 선택적으로 지정이 가능하다.

- Range: bytes=-5 : object의 마지막 5bytes
- Range: bytes=10-15 : 처음 10bytes 이후 5bytes
- Range: bytes=32- : 처음 32bytes 이후 전체 object
- If-Match
 - header에 지정한 Etag값이 실제 object의 Etag값과 일치할 경우, 200 OK를 전달하고, 일치하지 않을 경우 412 Precondition Failed 전달
- If-None-Match
 - header에 지정한 Etag값이 실제 object의 Etag값과 일치하지 않을 경우, 200 OK를 전달하고, 일치할 경우 304 Not Modified 전달
- If-Modified-Since
 - header에 지정한 날짜가 이후 수정된 object가 있으면 200 OK를 전달하고, 없으면 304 Not Modified 전달
- If-Unmodified-Since
 - header에 지정한 날짜가 이후 수정되지 않은 object가 있으면 200 OK를 전달하고, 있으면 412 Precondition Failed를 전달

Request Body

GET 스토리지 object API에서는 request body를 사용하지 않는다.

● Response

Response Headers

Header Name	Description
-------------	-------------

Last-Modified	최종 수정일
Etag	MD5 checksum
Content-type	response body의 형식
Content-Length	response body의 길이

Response Body

object data

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	200
인증 실패(token error)	401
권한 없음	403
요청한 object 존재하지 않음	404

- **Example**

Sample Request

```
GET /v1/<account>/<container>/<object> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 200 OK
Last-Modified: Tue, 05 Apr 2011 02:29:03 GMT
Etag: e1c38a7aa52f9efa061de5913416e54e
Content-Length: 2046
Content-Type: binary/octet-stream
Date: Tue, 05 Apr 2011 02:29:27 GMT
```

```
[.....object data.....]
```

5.3 PUT Storage object

- API 설명

object를 업로드한다.

object metadata 및 content data를 쓰기/덮어쓰기를 수행한다. 즉, 새로운 object를 업로드 하며, 기존에 있는 object를 업로드 할 경우 덮어쓰기가 수행된다.

또한, 사용자 metadata를 http header에 추가하여 수정이 가능하며, HEAD 스토리 object API를 통해서 response header로부터 조회가 가능하다. 요청헤더에 Content-Length 및 Content-Type이 누락되서는 안된다.

업로드 가능한 object의 최대 크기는 5GB이하 이며, 5GB를 넘는 object의 경우 **5.4 PUT Storage Large object**를 참조한다.

(주의) object 사용자 usermedata의 경우, 마지막으로 요청된 사용자 metadata가 저장된다. 즉, 이전에 저장된 사용자 metadata는 모두 삭제되고 마지막으로 요청된 사용자 metadata 가 저장된다. 이전 사용자 metadata를 유지하려면 모든 PUT 요청마다 header에 포함시켜야 한다. 기 저장된 파일에 대해 사용자 metadata 설정없이 PUT 요청할 경우 이전 모든 사용자 metadata는 삭제된다.

- Request Syntax

```
PUT /<api version>/<account>/<container>/<object> HTTP/1.1
```

```
Host : 스토리지 서비스 서버
```

```
X-Auth-Token : 인증 토큰
```

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
Content-Type	object의 형식	String	
Content-Length	object의 크기	integer	
X-Object-Meta-*	사용자 metadata (*는 임의의 string)	String	No

(참고) http Etag header를 사용해서 object에 대한 checksum을 제공하여 upload 시, data integrity를 확인할 수 있다. Etag header의 사용은 강제적인 것은 아니다. 만일 put request 요청시 Etag를 제공하지 않으면 response header에서 제공하는 Etag 값을 이용하여 클라이언트에서 로컬하게 MD5 checksum을 수행하여 data integrity를 확보할 수 있다.

Request Body

object data

- **Response**

Response Headers

Header Name	Description	비 고
Etag	object MD5 checksum	
Last-Modified	최종 수정일	
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	201
인증 실패(token error)	401
권한 없음	403
Content-Length	411
Etag 값 불일치(수행불가)	422

- **Example**

Sample Request

```
PUT /v1/<account>/<container>/<object> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Object-Meta-genre: romantic comedy'
X-Object-Meta-subject: human relation'
X-Object-Meta-location: Korea,America'
```

```
[.....object data.....]
```

Sample Response

```
HTTP/1.1 201 Created
Date: Thu, 07 Jun 2007 18:57:07 GMT
Etag: d9f5eb4bba4e2f2f046e54611bc8196b
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```


5.4 PUT Storage Large object

- API 설명

5GB 크기를 넘는 object를 업로드한다.

KT ucloud storage service에서 지원하는 최대 업로드 파일 크기는 5GB이다. 따라서, 5GB를 넘는 파일의 업로드 및 다운로드를 다루기 위해서 몇 가지 절차를 수행하면 된다.

- 업로드 방법 및 절차

5GB 크기를 넘는 파일을 업로드 하기 위해서 아래와 같은 준비 및 절차가 필요

- (1) 5GB 크기를 넘는 파일을 먼저 5GB 이하의 작은 크기로 분할한다. 이때, 분할의 크기 및 수량에는 제한이 없다. API에서 직접적인 분할을 지원하지는 않는다.
- (2) 작은 크기로 분할된 각 segment를 업로드 한다. 개별 segment를 업로드 할 경우, 반드시 동일한 container에 업로드 해야하고, object 이름에 동일한 prefix를 가져야 한다. 또한, 각 segment의 이름은 향후 다운로드 시 연결시키기 위해 순서적으로 정렬이 가능해야 한다.

개별 segment업로드는 기존 업로드 API를 활용하면 된다.

- (3) 마지막으로 manifest 파일을 업로드해야 한다. manifest 파일의 크기는 0이며, 반드시 개별 segment가 저장되어 있는 container에 업로드 되어야 할 필요는 없다. 이 manifest 파일은 segment에 위치 정보를 담고 있는 정보 파일로 볼 수 있다. 위 manifest 파일을 업로드 할 경우, 반드시 헤더 X-Object-Manifest를 지정해야 하며, 그 값은 <container>/<prefix>를 설정해야 한다.

(주의) container의 ACL을 공개(.r:*) 설정할 경우에는 manifest container가 공개 설정 되어 있어야 함은 물론, segments의 container 헤더 x-container-

read 에 .:* 및 .listings를 추가해야 한다.

자세한 사용 예는 example을 참고

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
Content-Length	request body의 크기(파일크기)	integer	

Request Body

object data

- **Response**

Response Headers

Header Name	Description	비 고
Etag	object MD5 checksum	
Last-Modified	최종 수정일	
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	201
인증 실패(token error)	401
권한 없음	403
Content-Length	411
Etag 값 불일치(수행불가)	422

- **Example**

Sample Request

(전제) 현재 test, test_segments 두개의 container가 있고, 9G 크기가 되는 khm.data 파일이 있다. khm.data 파일을 3GB 파일 3개로(1.data, 2.data, 3.data) 분할 했다. 각 segment는 test_segments에 업로드 하고, manifest 파일은 test에 업로드 한다.

첫번째 segment 업로드

```
PUT /v1/<account>/test_segments/khm.data/1.data HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

```
[.....object data.....]
```

두번째 segment 업로드

```
PUT /v1/<account>/test_segments/khm.data/2.data HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

```
[.....object data.....]
```

세번째 segment 업로드

```
PUT /v1/<account>/test_segments/khm.data/3.data HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

```
[.....object data.....]
```

manifest file 업로드

```
PUT /v1/<account>/test/khm.data HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Object-Manifest: test_segments/khm.data/
Content-Length: 0
```

원본 khm.data 파일을 다운로드 하기 위해서는 object GET API를 사용하여 manifest 파일을 다운로드하면, 분할된 segment들이 합체가 되어 하나의 khm.data 파일로 다운로드가 된다. 즉, 단순히 manifest 파일을 다운로드 하는 것은 아니다.

khm.data 다운로드

```
GET /v1/<account>/test/khm.data HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

container ACL 추가

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : 'X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
x-container-read: .r:*, .rlistings
```

5.5 Chunked Transfer Encoding

- API 설명

- object를 업로드한다.

- object를 업로드 하기 위해 먼저 크기를 헤더에 명시해야 한다. 그러나, 업로드 object의 크기를 모르더라도 업로드가 가능하다. HTTP header 에서 Transfer-Encoding: chunked 를 이용해서 가능하다.

- Request Syntax

```
PUT /<api version>/<account>/<container>/<object> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
Transfer-Encoding: chunked
```

Required Request Headers

Header Name	Description	Type	Required
-------------	-------------	------	----------

X-Auth-Token	Auth Token	String	Yes
Transfer-Encoding	chunked	String	
Etag	object MD5 checksum	hex value	No
X-Object-Meta-*	custom metadata	String	

Request Body

object data

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	201
인증 실패(token error)	401
권한 없음	403
Etag 값 불일치(수행불가)	422

5.6 Copy object

- **API 설명**

서버에서 **object**를 복사한다.

만일 잘못된 이름이나 파일형식으로 업로드 했거나 다른 컨테이너로 파일을 옮겨야 할 필요가 있는 경우, 기존 파일을 삭제하고 업로드를 다시해야만 한다.

그러나, 서버에서 copy 기능을 지원한다면 재차 업로드를 하지 않고도 위와 같은 필요한 작업을 손쉽게 할 수 있다.

본 API는 그러한 기능을 제공하는 것으로써, 새로운 object에 대한 PUT 요청을 하면서 요청 헤더('X-Copy-From')에 data의 source를 명시하면 된다. 앞에서 언급한 헤더의 값으로는 반드시 `"/container/object"` 형식을 가져야 하며, 비록 크기가 0일찌라도, Content-Length를 지정해야만 한다.

- **Request Syntax**

```
PUT /<api version>/<account>/<container>/<destobject> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : 인증토큰
X-Copy-From : /<container>/<sourceobject>
Content-Length: 0
```

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Copy-From	source object의 위치	String	
Content-Length	body의 길이	integer	

- HTTP Status Codes

Description	HTTP Status code
요청처리 완료	201
인증 실패(token error)	401
권한 없음	403
Content-Length 누락	411

5.7 POST Storage object : 사용자 metadata

- API 설명

object metadata를 추가한다.

'X-Object-Meta-' 접두어를 갖는 key/value 형태의 object metadata를 추가한다.

Etag와 같은 header는 수정 불가하다. object 사용자 metadata 삭제는 일괄 삭제만 가능하고 특정 key만 삭제는 불가능하다.

(주의) account 및 aontainer 사용자 metadata API와 달리, object 사용자 metadata는 마지막 POST 요청에 대한 사용자 metadata를 저장한다. 따라서, 이전 요청에 의해 추가된 object 사용자 metadata는 모두 삭제되고 마지막 POST요청에 의한 사용자 metadata가 저장된다. 기 추가한 사용자 metadata 유지를 원할 경우, 모든 POST요청에 기 추가한 사용자 metadata를 항상 포함시켜야 한다. 만 일 사용자 metadata를 설정하지 않은 POST 요청을 할 경우, 이전 모든 사용자 metadata는 일괄 삭제됨으로 주의가 필요하다.

- **Request Syntax**

POST /<api version>/<account>/<container>/<object> HTTP/1.1
 Host : 스토리지 서비스 서버
 X-Auth-Token : 인증 토큰
 X-Object-Meta-key : *value*

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Object-Meta-*	추가/변경하고자 하는 Metadata	String	Yes

(*) 임의의 String

Request Body

POST 스토리지 object API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Server	서비스 서버
Content-type	response body의 형식
Content-Length	response body의 길이

Response Body

처리결과에 대한 HTML문서

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	202
인증 실패(token error)	401
권한 없음	403
요청된 object 존재하지 않음	404

- **Example1**

Sample Request

```
POST /v1/<account>/<container>/<object> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Object-Meta-Fruit: Apple
X-Object-Meta-Veggie: Carrot
```

Sample Response

```
HTTP/1.1 202 Accepted
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

- **Example2**

Example1 에서 추가한 object metadata 를 일괄 삭제하는 예이다.

Sample Request

```
POST /v1/<account>/<container>/<object> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 202 Accepted
Date: Thu, 01 Nov 2012 09:07:11 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```


5.8 DELETE Storage object

- **API 설명**

object를 삭제한다.

영구적인 object 삭제 (metadata 및 content)

- **Request Syntax**

```
DELETE /<api version>/<account>/<container>/<object> HTTP/1.1
```

Host : 스토리지 서비스 서버

X-Auth-Token : 인증 토큰

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes

Request Body

없음

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식

Response Body

없음

- HTTP Status Codes

Description	HTTP Status code
삭제 처리 완료	204
인증 실패(token error)	401
권한 없음	403
요청된 object 존재하지 않음	404

- Example

Sample Request

```
DELETE /v1/<account>/<container>/<object> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Type: text/plain; charset=UTF-8
```

제 6 장 S3 Compatible API

6.1 GET Service

- API 설명

container 리스트를 조회한다.

전체 container에 대한 리스트를 조회한다.

- Request Syntax

```
GET / HTTP/1.1
```

```
Host : 스토리지 서비스 서버
```

```
Date: date
```

```
Authorization: signatureValue
```

```
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- Response

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

XML

- **HTTP Status Codes**

Description	HTTP Status code
처리 완료	200
잘못된 요청	400
권한 없음	401

- **Example**

Sample Request

```
GET / HTTP/1.1
Host: ssproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
Content-Length: 326
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Buckets> <Bucket> <Name>test</Name> <CreationDate>2009-02-
03T16:45:09.000Z</CreationDate> </Bucket>
  <Bucket> <Name>ttt</Name> <CreationDate>2009-02-
03T16:45:09.000Z</CreationDate> </Bucket> </Buckets>
</ListAllMyBucketsResult>
```

6.2 GET Bucket

- API 설명

container에 저장된 object 리스트를 조회한다.

특정 container에 저장되어 있는 object의 리스트를 조회한다.

- Request Syntax

GET / HTTP/1.1

Host : *containername*.스토리지 서비스 서버

Date: date

Authorization: signatureValue

Content-Type: application/x-www-form-urlencoded; charset=utf-8

Request Parameter

Parameter	Description	Required
max-keys	container조회 결과에 최대 출력개수 지정 Type : 정수값	No
marker	주어진 string값보다 큰 name을 가진 container 출력 Type : string	

prefix	주어진 string값으로 시작하는 object를 출력 Type : string	
delimiter	object를 그룹핑 하기 위해 사용하는 문자 Type : string	

Request Body

- **Response**

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

XML

- **HTTP Status Codes**

Description	HTTP Status code
처리 완료	200
잘못된 요청	400
권한 없음	401
container 없음	404

- **Example**

Sample Request

```
GET / HTTP/1.1
Host: test.ssproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
Content-Length: 355
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <Prefix> </Prefix>
  <Marker> </Marker>
  <Delimiter> </Delimiter>
  <IsTruncated> </IsTruncated>
  <MaxKeys> </MaxKeys>
  <Name>myobject</Name>
  <LastModified>2009-10-12T17:50:30.000Z</LastModified>
  <ETag>fba9dede5f27731c9771645a39863328 </ETag>
  <Size>434234</Size>
  <StorageClass>STANDARD</StorageClass> </Contents>
</ListBucketResult>
```

6.3 GET Object

- **API 설명**
object를 다운로드 한다.
- **Request Syntax**

```
GET /objectname HTTP/1.1
Host : containername.스토리지 서비스 서버
Date: date
Authorization: signatureValue
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- **Response**

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

- **HTTP Status Codes**

Description	HTTP Status code
처리 완료	200
잘못된 요청	400
권한 없음	401
object 없음	404

- **Example**

Sample Request

```
GET /myobject HTTP/1.1
Host: test.ssproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8
Content-Length: 355
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive
```

6.4 HEAD Object

- **API 설명**
object의 metadata를 조회한다.
- **Request Syntax**

```
HEAD /objectname HTTP/1.1
Host : containername.스토리지 서비스 서버
Date: date
Authorization: signatureValue
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- **Response**

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

- **HTTP Status Codes**

Description	HTTP Status code
처리 완료	200
잘못된 요청	400
권한 없음	401
object 없음	404

- **Example**

Sample Request

```
HEAD /myobject HTTP/1.1
Host: test.sproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 200 OK
x-amz-id-2:
gyB+3jRPnrkN98ZajxHXr3u7EFM67bNgSAxexeEHndCX/7GRnfTXxReKUQF28IfP
Last-Modified: Mon, 05 Mar 2012 17:53:54 GMT
ETag: "dfe4b5a232d391118e1d0c2fe7491fb2"
Date: Tue, 06 Mar 2012 01:57:38 GMT
Accept-Ranges: bytes
Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept x-amz-request-id: 1AH0DCE7E066B176
Content-Type: application/octet-stream; charset=UTF-8
Content-Length: 18
```

6.5 PUT Object

- **API 설명**

object를 업로드 한다.

- **Request Syntax**

```
PUT /objectname HTTP/1.1
Host : containername.스토리지 서비스 서버
Date: date
Authorization: signatureValue
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- **Response**

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

- **HTTP Status Codes**

Description	HTTP Status code
처리 완료	201
잘못된 요청	400
권한 없음	401
container 없음	404

- **Example**

Sample Request

```
PUT /myobject HTTP/1.1
Host: test.ssproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=UTF-8
Content-Length: 355
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive
```

6.6 PUT Bucket

- API 설명
container를 생성한다.
- Request Syntax

```
PUT / HTTP/1.1
Host : containername.스토리지 서비스 서버
Date: date
Authorization: signatureValue
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- Response

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

- **HTTP Status Codes**

Description	HTTP Status code
처리 완료	201
요청 container 존재	202
잘못된 요청	400
권한 없음	401

- **Example**

Sample Request

```

PUT / HTTP/1.1
Host: test.ssproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive

```

Sample Response

```

HTTP/1.1 201 Created
Content-Type: application/xml; charset=UTF-8
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive

```

6.7 DELETE Bucket

- **API 설명**
container를 삭제한다.
- **Request Syntax**

```
DELETE / HTTP/1.1
Host : containername.스토리지 서비스 서버
Date: date
Authorization: signatureValue
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- **Response**

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

- **HTTP Status Codes**

Description	HTTP Status code
삭제 완료	204
잘못된 요청	400
권한 없음	401
container 없음	404

- **Example**

Sample Request

```
DELETE / HTTP/1.1
Host: test.ssproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 204 No Content
Content-Type: application/xml; charset=UTF-8
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive
```

6.8 DELETE Object

- **API 설명**
object를 삭제한다.
- **Request Syntax**

```
DELETE /myobject HTTP/1.1
Host : containername.스토리지 서비스 서버
Date: date
Authorization: signatureValue
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- **Response**

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

- **HTTP Status Codes**

Description	HTTP Status code
삭제 완료	204
잘못된 요청	400
권한 없음	401
object 없음	404

- **Example**

Sample Request

```
DELETE /myobject HTTP/1.1
Host: test.ssproxy.ucloudbiz.olleh.com
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 204 No Content
Content-Type: application/xml; charset=UTF-8
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive
```

6.9 PUT Object Copy

- **API 설명**
object를 복사한다.

- **Request Syntax**

```
PUT/myobject HTTP/1.1  
Host : containername.스토리지 서비스 서버  
x-amz-copy-source: /source_container/sourceObject  
Date: date  
Authorization: signatureValue  
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

Request Parameter

Request Body

- **Response**

Response Headers

Header Name	Description	비 고
Content-type	response body의 형식	
Content-Length	response body의 길이	

Response Body

- **HTTP Status Codes**

Description	HTTP Status code
복사 성공	201
잘못된 요청	400
권한 없음	401
object 없음	404

- **Example**

Sample Request

```
PUT /myobject2 HTTP/1.1
Host: test2.ssexpoxucloudbiz.olleh.com
x-amz-copy-source: /test/myobject
Authorization: AWS test@test.com:6edfdDtNfIBUmamh0XNI/IHdr68=
Date: Thu, 03 May 2012 21:04:36 GMT
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Connection: Keep-Alive
```

Sample Response

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=UTF-8
X-Trans-Id: tx72d9a430fab54e438b9067c8b0c44127
Date: Thu, 03 May 2012 21:04:34 GMT
Connection: keep-alive
```

제 7 장 Reseller service.

Reseller service는 다수의 사용자를 한 계정에서 사용하도록 지원하는 서비스로서 포탈에서 신청 가능하다. API는 크게 2종류로서 reseller service 를 신청한 계정으로 인증을 받는 사용자 인증 API(7.1), 내부의 사용자를 관리하기 위한 API(7.2-7.6)가 있다. 이외의 API들은 공통적으로 사용한다. 추가적으로 container에 사용자가 사용할 수 있도록 권한을 부여하는 API(4.6)을 참고하면 된다.

7.1 사용자 인증

- API 설명

사용자 인증을 요청한다.

스토리지 서비스 사용을 위한 사용자 인증을 처리한다. 포탈에 입력한 계정 이름, admin 사용자 이름, 암호를 사용한다. 인증이 완료된 이후에는 response header를 통해서 storageUrl 및 token이 전달된다. 2.1 장의 사용자 인증과 비슷하며 인증시 필요한 내용만 차이가 있다.

- Request Syntax

```
GET /storage/v1/auth HTTP/1.1
Host : 인증 서버
X-Storage-User : 계정이름:사용자id
X-Storage-Pass : 사용자 패스워드
```

Request Parameters

사용자 인증 API에서는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
-------------	-------------	------	----------

host	인증서버	String	Yes
X-Storage-User	account:사용자id	String	
X-Storage-Pass	사용자 패스워드	String	

Request Body

사용자 인증 API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-Length	response body의 길이(bytes)
X-Storage-Url	스토리지 서비스 URL
X-Auth-Token	스토리지 서비스 사용 인증을 위한 토큰

Response Body

json 스트링 타입으로 요청에 대한 결과를 제공한다.

제공하는 json key는 storage이며, storageUrl을 담고 있다.

자세한 예는 example을 참조할 것.

- **HTTP Status Code**

Description	HTTP Status code
인증 완료	200
인증 실패	401

- **Examples**

Sample Request

```
GET /storage/v1/auth HTTP/1.1
Host : api.ucloudbiz.olleh.com
X-Storage-User : test:user
X-Storage-Pass : password
```

Sample Response

```
HTTP/1.1 200 OK
X-Storage-Url: https://ssproxy.ucloudbiz.olleh.com/v1/AUTH_024067c0-f236-4d2a-80d0-736698fb6d36
X-Storage-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
Content-Length: 112
Date: Mon, 21 Feb 2011 08:03:04 GMT
```

7.2 GET account

- **API 설명**

account 정보를 조회한다.

스토리지 서비스의 reseller service를 신청한 account에 대한 정보를 조회한다.
storageUrl, account_id, 소속 user에 대한 리스트를 볼 수 있다.

- **Request Syntax**

```
GET /auth/v2/account HTTP/1.1  
Host : 인증 서버  
X-Auth-Admin-User : 계정:관리자id  
X-Auth-Admin-key : 관리자 패스워드
```

Request Parameters

GET account API에서는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes
X-Auth-Admin-User	account:관리자id	String <account:user>	
X-Auth-Admin-Key	관리자 패스워드	String	

Request Body

인증 API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

Response Body

json 스트링 타입으로 요청에 대한 결과를 제공한다.

제공하는 json key는 services, account_id, users이고 각각은 서비스 url, 내부적으로 관리하는 account id 및 해당 account에 포함되어 있는 사용자를 나타내고 있다. 자세한 예는 example을 참조할 것.

- **HTTP Status Code**

Description	HTTP Status code
요청처리 성공	200
잘못된 요청	400
권한 없음	403
account 정보 없음	404

- **Examples**

Sample Request

```
GET /auth/v2/test HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Admin-User: test:user
X-Auth-Admin-Key: password
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 237
Date: Thu, 17 Feb 2011 07:06:24 GMT

{"services": {"storage": {"default": "local", "local":
  "https://ssproxy.ucloudbiz.olleh.com/v1/AUTH_024067c0-f236-4d2a-80d0-
  736698fb6d37"}}, "account_id": "AUTH_024067c0-f236-4d2a-80d0-
  736698fb6d37", "users": [{"name": "user"}, {"name": "user2"}]}
```

7.3 GET user

- **API 설명**

사용자 정보를 조회한다.

account에 소속된 사용자 정보를 조회한다.

- **Request Syntax**

```
GET /auth/v2/account/user_id HTTP/1.1
```

Host : 인증 서버

X-Auth-Admin-User : 계정:관리자id

X-Auth-Admin-key : 관리자 패스워드

Request Parameters

GET user API에서는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes
X-Auth-Admin-User	계정:관리자id	String <account:user>	
X-Auth-Admin-Key	관리자 패스워드	String	

Request Body

GET user API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
-------------	-------------

Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

Response Body

json 스트링 타입으로 요청에 대한 결과를 제공한다.

제공하는 json key는 groups, auth이고 각각 account, user_id 및 권한 정보와 패스워드 정보를 나타낸다. 자세한 예는 example을 참조할 것.

- **HTTP Status Code**

Description	HTTP Status code
요청완료	200
요청거절(권한없음)	403
해당 user_id 없음	404

- **Examples**

Sample Request

```
GET /auth/v2/test/user2 HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Admin-User: test:user
X-Auth-Admin-Key: password
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 86
Date: Mon, 21 Feb 2011 06:06:53 GMT
{"groups": [{"name": "test:user2"}, {"name": "test"}], "auth": "plaintext:password"}
```

7.4 GET users list

- **API 설명**

사용자 리스트를 조회한다.

account에 포함되어 있는 전체 사용자 정보 및 그룹을 조회한다.

- **Request Syntax**

```
GET /auth/v2/account/.groups HTTP/1.1
```

Host : 인증 서버

X-Auth-Admin-User : 계정:관리자id

X-Auth-Admin-key : 관리자 패스워드

Request Parameters

GET user API에서는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes
X-Auth-Admin-User	account:관리자id	String <account:id>	
X-Auth-Admin-Key	관리자 패스워드	String	

Request Body

GET user API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
-------------	-------------

Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

Response Body

json 스트링 타입으로 요청에 대한 결과를 제공한다.

제공하는 json key는 account이고 배열 형태로 해당 account에 포함된 user_id를 포함하고 있다. 자세한 예는 example을 참조할 것.

- **HTTP Status Code**

Description	HTTP Status code
요청완료	200
요청거절(권한없음)	403
해당 user_id list 없음	404

- **Examples**

Sample Request

```
GET /auth/v2/test/.groups HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Admin-User: test:user
X-Auth-Admin-Key: password
```

Sample Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 73
Date: Mon, 21 Feb 2011 06:22:51 GMT

{"accounts": [{"name": "test"}, {"name": ".admin"}, {"name": "test:user"}, {"name": "test:user2"}]}
```

7.5 PUT user

- **API 설명**

사용자를 생성한다.

account에 새로운 사용자를 생성한다. 기존에 동일한 사용자가 존재할 경우, 요청이 거절되지 않고 비밀번호 변경처리를 하거나 권한을 변경할 수 있다.

- **Request Syntax**

```
PUT /auth/v2/<계정>/<사용자> HTTP/1.1
Host : 인증 서버
X-Auth-Admin-User : 계정:관리자id
X-Auth-Admin-key : 관리자 비밀번호
X-Auth-User-Key : 생성할 사용자 비밀번호
```

Request Parameters

PUT user API에서는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes
X-Auth-Admin-User	account:관리자id	String <account:id>	
X-Auth-Admin-Key	관리자 비밀번호	String	
X-Auth-User-Key	생성할 사용자 비밀번호	String	

Optional Request Headers

Header Name	Description	Type	Value
-------------	-------------	------	-------

X-Auth-User-Admin	user_admin 생성	String	true/false
-------------------	---------------	--------	------------

위 요청 헤더를 사용하지 않을 경우, false 처리가 되고 단순 user 생성

Request Body

PUT user API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

Response Body

html 문서를 제공하며, 요청에 대한 결과를 문서의 head와 body에 표시한다.

- **HTTP Status Code**

Description	HTTP Status code
user_id 생성 완료	201
잘못된 요청	400
요청 거절(권한 없음)	403

- **Examples**

Sample Request

```
GET /auth/v2/test/user3 HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Admin-User: test:user
X-Auth-Admin-Key: password
X-Auth-User-Key: 123
```

Sample Response

```
HTTP/1.1 201 Created
Content-Length: 118
Content-Type: text/html; charset=UTF-8
Date: Mon, 21 Feb 2011 07:35:27 GMT
```

```
<html>
  <head>
    <title>201 Created</title>
  </head>
  <body>
    <h1>201 Created</h1>
    <br /> <br />
  </body>
</html>
```


7.6 DELETE user

- **API 설명**

사용자를 삭제한다.

account의 특정사용자를 삭제한다.

- **Request Syntax**

```
DELETE /auth/v2/<계정>/<사용자> HTTP/1.1
```

Host : 인증 서버

X-Auth-Admin-User : 계정:관리자id

X-Auth-Admin-key : 관리자 패스워드

Request Parameters

DELETE user API에서는 request parameter를 사용하지 않는다.

Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes
X-Auth-Admin-User	account:관리자id	String <account:id>	
X-Auth-Admin-Key	관리자 패스워드	String	

Request Body

DELETE user API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-Type	response body의 형식
Content-Length	response body의 길이(bytes)

Response Body

DELETE user API에서는 response body를 사용하지 않는다.

- **HTTP Status Code**

Description	HTTP Status code
user_id 삭제 완료	204
요청 거절(권한없음)	403
요청한 user_id 존재하지 않음	404

- **Examples**

Sample Request

```
DELETE /auth/v2/test/user3 HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Admin-User: test:user
X-Auth-Admin-Key: password
```

Sample Response

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Date: Mon, 21 Feb 2011 07:35:27 GMT
```

제 8 장 TempURL Service

TempURL service는 인증없이 파일 다운로드와 업로드를 할 수 있는 서비스이다. 오브젝트 단위로 TempURL을 생성할 수 있다. TempURL은 알고리즘을 통해 시그니처를 생성하고 유효 시간을 설정해야 한다. 해당 시간이 경과하면 더 이상 서비스를 이용할 수 없다.

8.1 account 설정

account 에 key 값 설정

account meta header에 POST 요청을 통해 key값을 설정한다.

설정된 key값은 시그니처 생성 시 사용된다.

POST 요청은 본 문서의 3.3 POST Storage account 에 설명되어 있다.

X-Account-Meta-Temp-URL-Key=*mykey*

8.2 시그니처 생성

시그니처 생성 알고리즘 (python)

```
import hmac
from hashlib import sha1
from time import time

method = 'GET'
expires = int(time() + 60)
```

```

path = '/v1/account/container/object'
key = 'mykey'
hmac_body = '%sWn%sWn%s' % (method, expires, path)
sig = hmac.new(key, hmac_body, sha1).hexdigest()

```

- * method : GET, PUT
- * expires : TempURL 유효 시간, 초 단위, the Unix timestamp 형식
- * account : account URL (예 : AUTH_123-456-789)
- * container : 파일 박스 이름
- * object : 파일 이름
- * key : account meta header 로 설정한 값 (X-Account-Meta-Temp-URL-Key=*mykey*)
- * hash 알고리즘 : HMAC-SHA1 (RFC 2104)

시그니처 생성 시 사용 method	요청 가능 method
GET	GET, HEAD
PUT	PUT, HEAD

8.3 TempURL 생성

시그니처와 유효 시간을 이용하여 TempURL 을 생성한다.

TempURL 에 필요한 파라미터

```

temp_url_sig
temp_url_expires

```

TempURL

https://ssproxy.ucloudbiz.olleh.com/v1/account/container/object?temp_url_sig=sig&temp_url_expires=expires

8.4 TempURL 요청

- GET Request Syntax

```
GET /v1/account/container/object?temp_url_sig=sig&temp_url_expires=expires  
HTTP/1.1  
Host : https://ssproxy.ucloudbiz.olleh.com
```

- **PUT Request Syntax**

```
GET /v1/account/container/object?temp_url_sig=sig&temp_url_expires=expires  
HTTP/1.1  
Host : https://ssproxy.ucloudbiz.olleh.com
```

Required Request Headers

Header Name	Description	Type	Required
Content-Type	object의 형식	String	yes
Content-Length	object의 크기	integer	

- **HEAD Request Syntax**

```
HEAD /v1/account/container/object?temp_url_sig=sig&temp_url_expires=expires  
HTTP/1.1  
Host : https://ssproxy.ucloudbiz.olleh.com
```

제 9 장 FormPost Service

FormPost는 web form에서 ucloud storage로 파일을 업로드할 수 있는 기능이다. 브라우저에서 보낸 form 요청을 object PUT 요청으로 변환해주며 5GB 이하의 파일만 업로드 가능하다.

9.1 account 설정

account에 key값 설정

account meta header에 POST 요청을 통해 key값을 설정한다.

TempURL과 동일한 헤더를 사용하고 설정한 key값은 시그니처 생성 시 사용된다.

POST 요청은 본 문서의 3.3 POST Storage account 에 설명되어 있다.

```
X-Account-Meta-Temp-URL-Key = mykey
```

9.2 시그니처 생성

시그니처 생성 알고리즘 (python)

```
import hmac
from hashlib import sha1
from time import time

path = '/v1/account/container/object_prefix'
redirect = 'http://mydomain.co/some-page' # set to '' if redirect not in form
max_file_size = 104857600
```

```

max_file_count = 10
expires = int(time() + 600)
key = 'mykey'
hmac_body = '%s\\n%s\\n%s\\n%s\\n%s' % (path, redirect,
                                     max_file_size, max_file_count, expires)
signature = hmac.new(key, hmac_body, sha1).hexdigest()

```

- * path : 오브젝트가 저장될 위치
- * account : account URL (예 : AUTH_123-456-789)
- * container : 파일 박스 이름
- * object_prefix : 파일 이름의 prefix
- * expires : FormPost 시그니처 유효 시간, 초 단위, the Unix timestamp 형식
- * key : account meta header 로 설정한 값 (X-Account-Meta-Temp-URL-Key=*mykey*)
- * hash 알고리즘 : HMAC-SHA1 (RFC 2104)
- ** expires 값과 signature 값을 FormPost 요청에 사용

9.3 FormPost 요청

```

<form action="<storage-url>" method="POST"
      enctype="multipart/form-data">
  <input type="hidden" name="redirect" value="<redirect-url>" />
  <input type="hidden" name="max_file_size" value="<bytes>" />
  <input type="hidden" name="max_file_count" value="<count>" />
  <input type="hidden" name="expires" value="<unix-timestamp>" />
  <input type="hidden" name="signature" value="<hmac>" />
  <input type="file" name="<file_name>" /> <br />
  <input type="submit" />
</form>

```

입력 값 설명

<storage-url>

업로드한 파일이 저장될 URL, object_prefix는 생략 가능

https://ssproxy.ucloudbiz.olleh.com/v1/AUTH_account/container/object_prefix

<redirect-url>

업로드 완료 후 응답을 받는 주소, 사용하지 않으면 공백 입력

<bytes>

최대 업로드 파일 크기 설정, MAX=5GB

<count>

해당 form을 이용하여 업로드할 수 있는 최대 파일 개수

<uix-timestamp>

시그니처 만료 시간, 해당 시간까지 시그니처 이용 가능

<hmac>

위의 알고리즘으로 계산된 시그니처 값

<file_name>

업로드 파일을 저장할 이름

이용 예

```
<form action="https://ssproxy.ucloudbiz.olleh.com/v1/AUTH\_497c8e6f-74b8-4034-b63c-cb78d6c91194/test\_formpost/" method="POST"
enctype="multipart/form-data">
    <input type="hidden" name="redirect" value="" />
    <input type="hidden" name="max_file_size" value="104857600" />
    <input type="hidden" name="max_file_count" value="10" />
    <input type="hidden" name="expires" value="1385109675" />
    <input type="hidden" name="signature"
value="3365349f18517cc42c82610274f9d0fc241c9431" />
    <input type="file" name="testfile" /> <br />
    <input type="submit" />
</form>
```


제 10 장 Expiring Objects

Object 단위로 삭제 스케줄을 설정할 수 있는 서비스 이다. Object를 일정 기간 동안만 저장하고자 할 때 유용하게 사용될 수 있다. Object PUT 혹은 POST 요청 시 X-Delete-At, X-Delete-After 헤더를 통하여 설정한다. Expiring Objects 에서는 초 단위로 시간을 입력하고 시간 포맷은 Unix Epoch timestamp (integer) 이다.

* 주의 사항 : Expiring Objects을 통하여 삭제된 파일은 복구할 수 없습니다. 정확한 설정과 확인을 통하여 불필요한 파일이 삭제되지 않도록 주의하시기 바랍니다.

10.1 Expiring Objects 설정

- 요청 유형

Method	Description	API 규격
PUT	Object 업로드 시에 Expiring Objects 설정	본문 5.3장
POST	기 업로드된 Object에 Expiring Objects 설정	본문 5.7장

- 헤더 유형

Method	Description	Type
X-Delete-At	Object가 삭제될 시간	integer
X-Delete-After	Object가 저장되는 기간	

- X-Delete-At

Sample Request

```
POST /v1/account/container/object HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Delete-At: 1339429105
```

- **X-Delete-After**

Sample Request

```
POST /v1/account/container/object HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Delete-After: 300
```

- **Expiring objects 설정 조회**

X-Delete-At 과 X-Delete-After 헤더를 통하여 Expiring objects 설정을 하면 모두 X-Delete-At 헤더로 Object 메타 헤더에 (변환)저장된다.

Sample Request

```
HEAD /v1/account/container/object HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 07 Jun 2007 18:57:07 GMT
Etag: 8a964ee2a5e88be344f36c22562a6486
Content-Length: 5120
Content-Type: image/jpeg
X-Delete-At: 1339429105
```

10.2 Expiring Objects 설정 해제

X-Remove-Delete 접두사를 통하여 Expiring Objects 설정을 해제할 수 있다.

- **X-Remove-Delete-At**

Sample Request

```
POST /v1/account/container/object HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Remove-Delete-At: 1339429105
```

- **X-Remove-Delete-After**

Sample Request

```
POST /v1/account/container/object HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Remove-Delete-After: 300
```

제 11 장 CORS(Cross Origin Resource Sharing)

이 설정을 통해 서로 다른 도메인 간 통신이 가능하다. Swift는 저장 되어있는 데이터에 대해 CORS 헤더 설정이 가능하며, container 단위로 헤더를 설정한다. X-Container-Meta-Access-Control-Allow-Origin 헤더를 container에 설정할 경우, 해당 container에 있는 object에 적용 된다.

11.1 CORS 메타데이터 설정

- 요청 유형

Method	Description	API 규격
PUT	Container 생성 시에 CORS 설정	본문 5.3장
POST	존재하는 Container에 CORS 설정	본문 5.7장

CORS 메타데이터를 추가한다.

Sample Request

```
PUT /v1/account/container HTTP/1.1
Host : sproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Container-Meta-Access-Control-Allow-Origin: http://kt-dev-test.net (원본 웹 서버)
```

11.2 CORS 요청하기

- CORS Pre-flight Request

응답이 200 OK 여야 하며, 이 외에 모든 응답의 종류는 요청 비정상으로 본다.

Sample Request

```
OPTIONS /v1/account/container/object HTTP/1.1
Host : sproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Origin: http://kt-dev-test.net
Access-Control-Request-Method: POST
```

Sample Response

```
HTTP/1.1 200 OK
Date: Thu, 07 Jun 2007 18:57:07 GMT
Access-Control-Allow-Origin: http://kt-dev-test.net
Access-Control-Allow-Methods: HEAD, GET, PUT, POST, COPY, OPTIONS, DELETE
Access-Control-Allow-Headers: x-auth-token
Allow: HEAD, GET, PUT, POST, COPY, OPTIONS, DELETE
Content-Length: 0
```

11.3 CORS 실제 요청

위의 요청에 대한 응답이 200 OK일 경우에만 실제 요청을 할 수 있다.

- CORS 실제 Request

Sample Request

```
POST /v1/account/container/object HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Origin: http://kt-dev-test.net
Content-Type: text/plain
```

Sample Response

```
HTTP/1.1 202 Accepted
Date: Thu, 07 Jun 2007 18:57:07 GMT
Access-Control-Allow-Origin: http://kt-dev-test.net
Content-Type: text/html; charset=UTF-8
Content-Length: 76
Access-Control-Expose-Headers: cache-control, content-language, content-type, expires, last-modified, pragma, etag, x-timestamp, x-trans-id
```

제 12 장 Cloudberry 사용자를 위한 API 설정

이 장은 cloudberry로 파일을 업로드 한 사용자를 위해 작성되었다. Cloudberry를 사용하여 파일을 업로드 하는 경우에는, 기본적으로 10MB 단위로 분할되어 업로드 된다. (분할 크기 설정은 cloudberry의 option - Advanced - Use chunks 에서 가능하며, Chunk transparency 체크 박스를 해제하면 분할된 파일들을 볼 수 있다.) 이 때, cloudberry와 swift API의 파일 분할 업로드 방식에 차이가 존재한다. 아래의 [example]을 참고한다.

(swift API는 기본적으로 5GB 단위로 파일을 분할하여 업로드하고, 분할 크기 설정도 가능하다.)

- **Example**

container 이름: Con / 파일 이름: test01

Swift API	Cloudberry
Con ----- test01 Con_segments --- test01_00001 --- test01_00002 --- test01_00003	Con ----- test01 ----- test01!CB00001 ----- test01!CB00002 ----- test01!CB00003

- 즉 Swift API를 사용하여 con 이라는 container에 "test01"이라는 파일을 분할 업로드하는 경우 Con_segments라는 container가 자동으로 생성이 되며, 해당 container(Con_segments)에 실제 파일이 분할되어 저장된다. Container(Con)의 "test01"은 분할된 파일들을 가리키는 0byte의 파일(manifest 파일)이다.
- Cloudberry를 사용하여 con 이라는 container에 test01이라는 파일을 분할 업로드하는 경우 새로운 container는 생성되지 않고, 기존 Container(Con)에 "test01" 파일이 "test01!CB00001, ..." 과 같이 분할되어 업로드 된다. Container(Con)의 "test01"은 분할된 파일들을 가리키는 0byte의 파일(manifest파일) 이다.

12.1 POST storage container : ACL

- **API 설명**

container의 ACL을 설정한다.

cloudberry로 파일을 분할하여 업로드 한 경우(사용자가 의도하지 않아도 기본적으로 10MB로 분할되어 업로드 되어 진다) cloudberry에서 container에 대한 공개 권

한 설정이 적용되지 않는다. 때문에, API를 미용하여 ACL 설정이 필요하다.

5.4 PUT Storage Large Object 하단의 (**주의**) 부분과 비슷하지만 cloudberry의 경우 실제 파일이 들어가는 container_segments가 생성되지 않으므로, 차이가 있다. Cloudberry로 업로드 한 파일이 있는 container의 ACL을 공개(.r:*) 설정할 경우에는 해당 container 헤더 x-contianer-read 에 .r:* 및 .rlistings를 추가해야 한다.

Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
x-container-read/write	ACL을 설정하기 위한 http header key	String	Yes

Request Body

POST 스토리지 container API에서는 request body를 사용하지 않는다.

- **Response**

Response Headers

Header Name	Description
Content-type	response body의 형식
Content-Length	response body의 길이

Response Body

없음

- **HTTP Status Codes**

Description	HTTP Status code
요청처리 완료	204
인증 실패(token error)	401
권한 없음	403

- **Example**

Sample Request

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy.ucloudbiz.olleh.com
X-Auth-Token : 'X-Auth-Token: AUTH_tk5f6d351c490b44b8b60b015e744a435a
x-container-read: .r:*,.rlistings
```

Sample Response

제 13

```
HTTP/1.1 204 No Content
Date: Thu, 07 Jun 2007 18:57:07 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

13.1 사용자 Token 및 서비스 URL 획득하기

SS서비스를 사용하기 위해서는 먼저 사용자 Token 및 서비스 URL를 요청/획득 과정을 반드시 거쳐야 한다. 이를 위해서는 제공받은 ID와 패스워드를 가지고 인증 요청을 수행해야 한다.

아래 예에서는 사용자 ID는 test:tester 이고 패스워드는 testing 이다.

- 사용자 인증 API 호출 : GET /auth/v1.0 HTTP1.1

요청헤더 : 'X-Storage-User: test:tester', 'X-Storage-Pass: testing'

- 응답헤더에서 사용자 Token 및 서비스 URL 가져오기

응답헤더 : 'X-Storage-Url:https://ssproxy.ucloudbiz.olleh.com/v1/AUTH_b54c7309-4fe2-496d-aeb4-496a564017a8'

'X-Storage-Token: AUTH_tk938f7f8746974c91b2992faa95ddca73'

13.2 일반적인 HTTP 요청

일단, 성공적으로 인증과정을 거쳐서 사용자 토큰 및 서비스 URL을 획득한 경우 Container 및 object에 대한 API를 호출하여 필요한 업로드, 다운로드, 생성 및 삭제 작업을 수행할 수 있다. 요청 헤더에는 반드시 사용자 Token를 명시해야 한다.

개별 API 마다 요청의 요구조건이 상이할 수 있으므로, 자세한 내용은 API 내용참조

- 사용자 인증 API 호출 : GET or PUT or POST or DELETE /v1/AUTH_b54c7309-4fe2-496d-aeb4-496a564017a8 HTTP1.1

요청헤더 : 'X-Auth-Token: AUTH_tk938f7f8746974c91b2992faa95ddca73'

- HTTP 응답코드, header 및 body를 해석하여 적절한 서비스 제공
구체적인 내용은 API 상세 내용 참조

13.3 인증요청 자바 webclient sample code

```
private boolean getAuth() {
    DefaultHttpClient hClient = new DefaultHttpClient();
    hClient.getConnectionManager().getSchemeRegistry().register(sch);

    HttpGet httpget = new HttpGet("https://ssproxy.ucloudbiz.olleh.com/auth/v1.0");
    httpget.setHeader("X-Storage-User", "test:tester");
    httpget.setHeader("X-Storage-Pass", "testing");
    HttpResponse response = null;
    HttpEntity entity = null;
    StatusLine res_sl = null;
    Header[] res_header = null;
    BufferedReader in = null;
    int returnCode = 0;
    boolean t = true;

    try {
        response = hClient.execute(httpget);
        res_sl = response.getStatusLine();
        returnCode = res_sl.getStatusCode();
        entity = response.getEntity();
        res_header = response.getAllHeaders();

        if(returnCode == 200) {
            for(int i=0; i < res_header.length; ++i) {
                if(res_header[i].getName().equals("X-Auth-Token"))
                    System.out.println(res_header[i].getValue());
                if(res_header[i].getName().equals("X-Storage-Url"))
                    System.out.println(res_header[i].getValue());
            }
        }
    }
}
```

```
        else { t = false; }  
    }  
    catch(IOException e) { e.printStackTrace(); }  
    hClient.getConnectionManager().shutdown();  
    return t;  
}
```