

---

# **KT ucloud storage2.0 service API**

**2016. 12**

---

## 문서 버전 및 이력

버전	일자	이력사항
v1	2016.12	최초 작성

## 목 차

<b>제 1 장 개요</b> .....	5
1.1 KT ucloud storage 2.0 service API 개요.....	5
1.2 ucloud storage 2.0 service API 기능 개요.....	6
<b>제 2 장 사용자 인증</b> .....	7
2.1 사용자 인증.....	7
<b>제 3 장 Storage account service</b> .....	13
3.1 GET Storage account.....	13
3.2 HEAD Storage account.....	18
3.3 POST Storage account : 사용자 metadata.....	20
<b>제 4 장 Storage container service</b> .....	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.....	37
4.7 POST Storage container static website.....	39
<b>제 5 장 Storage object service</b> .....	42
5.1 HEAD Storage object.....	42
5.2 GET Storage object.....	43
5.3 PUT Storage object.....	46
5.4 Chunked Transfer Encoding.....	48
5.5 Copy object.....	49
5.6 POST Storage object : 사용자 metadata.....	50
5.7 DELETE Storage object.....	52
<b>제 6 장 Dynamic Large Objects(DLO)</b> .....	53
<b>제 7 장 Static Large Objects(SLO)</b> .....	57
<b>제 8 장 TempURL Service</b> .....	60
8.1 account 설정.....	60
8.2 시그니처 생성.....	61
8.3 TempURL 생성.....	62
8.4 TempURL 요청.....	62
<b>제 9 장 FormPost Service</b> .....	63
9.1 account 설정.....	63

9.2 시그니처 생성 .....	64
9.3 FormPost 요청 .....	65
<b>제 10 장 Expiring Objects</b> .....	66
10.1 Expiring Objects 설정 .....	67
10.2 Expiring Objects 설정 해제 .....	68
<b>제 11 장 CORS(Cross Origin Resource Sharing)</b> .....	69
11.1 CORS 메타데이터 설정 .....	69
11.2 CORS 요청하기 .....	70
11.3 CORS 실제 요청 .....	71

## 제 1 장 개요

본 문서는 KT ucloud storage2.0 service 의 API 규격을 제시한다.

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

### 1.1 KT ucloud storage 2.0 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 2.0 service API 기능 개요

- 사용자 인증

기능	URI	설명
GET	<i>/v3/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

- 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

## 제 2 장 사용자 인증

### 2.1 사용자 인증

- API 설명

사용자 인증을 요청한다.

Keystone의 사용자 인증도 포탈에 등록된 email주소와 포탈로부터 받은 API Key를 사용한다. 인증이 완료되면 token이 사용자에게 전달 된다. token은 {"expires\_at": "YYYY-MM-DDT00:00:00"} 만료시간과 함께 전달 되며, token이 만료 시점을 지나면 401 에러를 리턴하므로 재발급 받아야 한다. Keystone 인증을 사용할 때에는 요청 body 부분에 사용자의 정보를 넣어줘야 한다.

ucloud storage2.0 스토리지 상품에 대한 인증 서버 URL은 아래와 같다.

KOR-Center	https://ssproxy2.ucloudbiz.olleh.com:5000/v3/auth
------------	---

- Request Syntax

```
POST /v3/auth/tokens HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
Content-Type: application/json
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "domain": {
            "id": "domain_ID"
          },
          "name": "Email_Address",
          "password": "user_API_Key"
        }
      }
    }
  },
}
```

```

    "scope": {
      "project": {
        "id": "project_id"
      }
    }
  }
}

```

### Required Request Headers

Header Name	Description	Type	Required
host	인증서버	String	Yes
Content-Type	application/json 헤더 설정	String	Yes

### Request Body

json 스트링 타입으로 요청한다.

Name	Description	Type	Required
domain	사용자의 도메인을 사용	object	Yes
auth	인증 object	object	Yes
name	ucloudbiz 사용자의 메일 주소 입력	string	Yes
user	사용자 object	object	Yes
password	ucloudbiz 사용자의 비밀번호 입력	string	Yes
identity	Identity object	object	Yes
methods	인증 method로, password 인증이기 때문에 password로 정의	array	Yes
scope	인증을 위한 scope object	object	Yes

- **Response**

### Response Headers

Header Name	Description
X-Subject-Token	스토리지 서비스 사용 인증을 위한 토큰
Content-Type	response body의 유형
Content-Length	response body의 길이(bytes)
X-Openstack-Request-Id	요청에 대한 ID



### Response Body

json 스트링 타입으로 요청에 대한 결과를 제공한다.

Name	Description	Type	Required
Token	Token object	object	Yes
audit_ids	Audit ID로 랜덤으로 생성되는 고유 값이며 토큰을 추적하는데 사용 가능	array	Yes
expires_at	인증 토큰의 만료 시점	string	Yes
issued_at	토큰이 발행되는 시점	string	Yes
Methods	인증 method로, password 인증일 때는 password로 정의	array	Yes
User	User object	object	Yes
Domain	Domain object	object	Yes
Id	사용자의 ID로 사용자의 name이나 id가 필요	String	No
Name	사용자의 name로 사용자의 name이나 id가 필요	String	No

- HTTP Status Code

Description	HTTP Status code
인증 완료	200
인증 실패	401
요청 형태 오류	400

- Examples : 인증요청

#### Sample Request

```
POST /v3/auth/tokens HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com:5000
Content-Type: application/json
{ "auth": {
  "identity": {
    "methods": ["password"],
    "password": {
      "user": {
        "name": "test@test.com",
```

```
        "domain": { "id": "2ff84e7139982ff8473ff8473" },
        "password": "Fi9ejS291Gsjfie3aSe0thjaB"
    }
}
},
"scope": {
    "project": {
        "id": "c49abe713998291Gsjf 73ff8473",
        "domain": { "id": "2ff84e7139982ff8473ff8473" }
    }
}
}
}
```

### Sample Response

```
HTTP/1.1 201 Created
Date: Sun, 18 Dec 2016 07:49:43 GMT
Server: Apache/2.4.18 (Ubuntu)
X-Subject-Token: e194e724dfb63a7194e724dfb63a3057
Vary: X-Auth-Token
X-Distribution: Ubuntu
x-openstack-request-id: req-5503dc20-3201-4066-9bbd-73fd17b0a9a6
Content-Length: 1867
Content-Type: application/json

{
  "token": {
    "audit_ids": [
      "fAkE3G1KucF21GWA0SBtw"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "id": "68fd904b4c0bc397526bca6904bbce79",
            "interface": "admin",
```

```

        "region": "RegionOne",
        "region_id": "RegionOne",
        "url": "http://ssproxy2.ucloudbiz.olleh.com:35357/v3"
    },
    {
        "id": "6704560a6904bbda2f1d29ae324c6d4f2",
        "interface": "internal",
        "region": "RegionOne",
        "region_id": "RegionOne",
        "url": "http://ssproxy2.ucloudbiz.olleh.com:5000/v3"
    },
    {
        "id": "6a6904bb8fd904b4c0bc397526bcce79",
        "interface": "public",
        "region": "RegionOne",
        "region_id": "RegionOne",
        "url": "http://ssproxy2.ucloudbiz.olleh.com:5000/v3"
    }
],
"id": "6fie732mr89a6904bb08eda5c7150cb146a",
"name": "keystone",
"type": "identity"
},
{
    "endpoints": [
        {
            "id": "6904bb08eda5c7150cb10e8597fd945c",
            "interface": "admin",
            "region": "RegionOne",
            "region_id": "RegionOne",
            "url":
"http://ssproxy2.ucloudbiz.olleh.com/v1/AUTH_6904bb08eda5c7150cb10e8597fd945
c "
        },
        {
            "id": "20506e44d1a74b05f19cb0cc74ce4b82",

```

```

        "interface": "internal",
        "region": "RegionOne",
        "region_id": "RegionOne",
        "url":
"http://ssproxy2.ucloudbiz.olleh.com/v1/AUTH_6904bb08eda5c7150cb10e8597fd945
c "
    },
    {
        "id": "fe228c96cd 30abaa09974a8099f5b803",
        "interface": "public",
        "region": "RegionOne",
        "region_id": "RegionOne",
        "url":
"http://ssproxy2.ucloudbiz.olleh.com/v1/AUTH_6904bb08eda5c7150cb10e8597fd945
c "
    }
],
    "id": "86904bb08eda5c715d945c b07fe604a6d",
    "name": "swift",
    "type": "object-store"
}
],
"expires_at": "2016-12-18T08:50:05.796158Z",
"issued_at": "2016-12-18T07:50:05.796182Z",
"methods": [
    "password"
],
"project": {
    "domain": {
        "id": "6904bb08eda58597fd945c 6651a1307",
        "name": "leeseul"
    },
    "id": "6904bb08eda5c7150cb10e8597fd945c ",
    "name": "leeseul"
},
"roles": [

```

```

    {
        "id": "ab62a2e4f6ba7a8a5c87862b680c6b8e4c",
        "name": "adminuser"
    }
],
"user": {
    "domain": {
        "id": "6904bb08eda5c7150cb10e8597fd945c ",
        "name": "leeseul"
    },
    "id": "99209dd585a73998c6a9e2eb74da7678",
    "name": "leeseul"
}
}
}

```

## 제 3 장 Storage account service

### 3.1 GET Storage account

- API 설명

**container** 리스트를 조회한다.

인증 API로부터 받은 Account 정보나 storage URL 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : 인증 토큰

```

#### Request Parameters

Parameter	Description	Required
Limit	container조회 결과에 최대 출력개수 지정 Type : 정수값	No

Marker	주어진 string값보다 큰 name을 가진 container 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)
X-Account-Storage-Policy-Policy- "policy이름"-Bytes-Used	주어진 storage policy에 저장된 스토리지 사용량(bytes)
X-Account-Storage-Policy-Policy- "policy이름"-Container-Count	주어진 storage policy에 저장된 container의 개수
X-Account-Storage-Policy-Policy- "policy이름"-Object-Count	주어진 storage policy에 저장된 object의 개수

#### 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : 5f6d351c490b44b8b60b015e744a435a
```

**Sample Response**

```
HTTP/1.1 200 OK
Content-Length: 48
Accept-Ranges: bytes
X-Account-Object-Count: 25
X-Account-Storage-Policy-Erasurecode-Container-Count: 1
X-Account-Storage-Policy-Policy-0-Container-Count: 4
X-Account-Storage-Policy-Policy-0-Bytes-Used: 86447
X-Account-Storage-Policy-Erasurecode-Bytes-Used: 239
X-Timestamp: 1479714696.55854
X-Account-Storage-Policy-Policy-0-Object-Count: 24
X-Account-Bytes-Used: 86686
X-Account-Container-Count: 5
Content-Type: text/plain; charset=utf-8
X-Account-Storage-Policy-Erasurecode-Object-Count: 1
x-account-project-domain-id: _unknown
X-Trans-Id: tx6164787b7dc6c3dce4f7d-005840c2fd
Date: Fri, 01 Dec 2016 00:40:29 GMT
```

```
con01
con02
ec01
swiftcon01
swiftcon01_segments
```

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

**Sample Request**

```
GET /v1/<account>?format=json HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
HTTP/1.1 200 OK
Content-Length: 247
Accept-Ranges: bytes
X-Account-Object-Count: 25
X-Account-Storage-Policy-Erasurecode-Container-Count: 1
X-Account-Storage-Policy-Policy-0-Container-Count: 4
X-Account-Storage-Policy-Policy-0-Bytes-Used: 86447
X-Account-Storage-Policy-Erasurecode-Bytes-Used: 239
X-Timestamp: 1479714696.55854
X-Account-Storage-Policy-Policy-0-Object-Count: 24
X-Account-Bytes-Used: 86686
X-Account-Container-Count: 5
Content-Type: application/json; charset=utf-8
X-Account-Storage-Policy-Erasurecode-Object-Count: 1
x-account-project-domain-id: _unknown
X-Trans-Id: txd5d68878e95d463a8a978-005840c34e
Date: Fri, 02 Dec 2016 00:41:50 GMT
```

```
[{"count": 12, "bytes": 75820, "name": "con01"}, {"count": 1, "bytes": 381, "name": "con02"}, {"count": 1, "bytes": 239, "name": "ec01"}, {"count": 1, "bytes": 0, "name": "swiftcon01"}, {"count": 10, "bytes": 10246, "name": "swiftcon01_segments"}]
```

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

**Sample Request**

```
GET /v1/<account>?format=xml HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

**Sample Response**



```
HTTP/1.1 200 OK
Content-Length: 506
Accept-Ranges: bytes
X-Account-Object-Count: 25
X-Account-Storage-Policy-Erasurcode-Container-Count: 1
X-Account-Storage-Policy-Policy-0-Container-Count: 4
X-Account-Storage-Policy-Policy-0-Bytes-Used: 86447
X-Account-Storage-Policy-Erasurcode-Bytes-Used: 239
X-Timestamp: 1479714696.55854
X-Account-Storage-Policy-Policy-0-Object-Count: 24
X-Account-Bytes-Used: 86686
X-Account-Container-Count: 5
Content-Type: application/xml; charset=utf-8
X-Account-Storage-Policy-Erasurcode-Object-Count: 1
x-account-project-domain-id: _unknown
X-Trans-Id: txa8f0367dd04045daa1cee-005840c419
Date: Fri, 02 Dec 2016 00:45:13 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<account name="AUTH_7d1b4bb2fc4b49f2b741f9585ce3f890">
  <container <name>con01 </name> <count>12 </count> <bytes>75820 </bytes> </c
ontainer>
  <container <name>con02 </name> <count>1 </count> <bytes>381 </bytes> </cont
ainer>
  <container <name>ec01 </name> <count>1 </count> <bytes>239 </bytes> </contai
ner>
  <container <name>swiftcon01 </name> <count>1 </count> <bytes>0 </bytes> </co
ntainer>
  <container <name>swiftcon01_segments </name> <count>10 </count> <bytes>102
46 </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 : ssproxy2.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 metadata**를 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

HEAD스토리지 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
X-Account-Storage-Policy-Policy- "policy이름"-Bytes-Used	주어진 storage policy에 저장된 스토리지 사용량(bytes)
X-Account-Storage-Policy-Policy- "policy이름"-Container-Count	주어진 storage policy에 저장된 container의 개수
X-Account-Storage-Policy-Policy- "policy이름"-Object-Count	주어진 storage policy에 저장된 object의 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

**Sample Response**

```
HTTP/1.1 204 No Content
Content-Length: 0
Accept-Ranges: bytes
```

```

X-Account-Object-Count: 25
X-Account-Storage-Policy-Erasurecode-Container-Count: 1
X-Account-Storage-Policy-Policy-0-Container-Count: 4
X-Account-Storage-Policy-Policy-0-Bytes-Used: 86447
X-Account-Storage-Policy-Erasurecode-Bytes-Used: 239
X-Timestamp: 1479714696.55854
X-Account-Storage-Policy-Policy-0-Object-Count: 24
X-Account-Bytes-Used: 86686
X-Account-Container-Count: 5
Content-Type: text/plain; charset=utf-8
X-Account-Storage-Policy-Erasurecode-Object-Count: 1
x-account-project-domain-id: _unknown
X-Trans-Id: txb22e022ed86d41679cd71-005840c4b3
Date: Fri, 02 Dec 2016 00:47:47 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 사용에는 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Account-Meta-Fruit: Test1
X-Account-Meta-key: metatest
```

#### Sample Response

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txe0b232b5459e4cea89e17-005840c693
Date: Fri, 02 Dec 2016 00:55:47 GMT
```

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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Remove-Account-Meta-Fruit: T
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)
X-Storage-Policy	Container의 policy 타입

**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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

- **Sample Response**

```
HTTP/1.1 204 No Content
Content-Length: 0
X-Container-Object-Count: 12
Accept-Ranges: bytes
X-Storage-Policy: Policy-0
X-Container-Bytes-Used: 75820
X-Timestamp: 1479714696.59461
Content-Type: text/plain; charset=utf-8
X-Trans-Id: tx45d66f55a629479fa629a-005840c7aa
Date: Fri, 02 Dec 2016 01:00:26 GMT
```

## 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 Type : 정수값	No



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)
X-Storage-Policy	Container의 policy 타입

### 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

**Sample Response**

```
HTTP/1.1 200 OK
Content-Length: 163
X-Container-Object-Count: 6
Accept-Ranges: bytes
X-Storage-Policy: Policy-0
X-Container-Bytes-Used: 47981
X-Timestamp: 1479714696.59461
Content-Type: text/plain; charset=utf-8
X-Trans-Id: txae fd3784c87f4f208d45c-005840c902
Date: Fri, 02 Dec 2016 01:06:10 GMT
```

```
java_docs/constant-values.html
java_docs/deprecated-list.html
java_docs/help-doc.html
java_docs/index.html
java_docs/serialized-form.html
java_docs/stylesheet.css
```

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

**Sample Request**

```
GET /v1/<account>/<container>?format=json HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
Content-Length: 0
```

X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb

### Sample Response

HTTP/1.1 200 OK  
Content-Length: 1051  
X-Container-Object-Count: 6  
Accept-Ranges: bytes  
X-Storage-Policy: Policy-0  
X-Container-Bytes-Used: 47981  
X-Timestamp: 1479714696.59461  
Content-Type: application/json; charset=utf-8  
X-Trans-Id: txecfb85b6d3f74550b73b2-005840c944  
Date: Fri, 02 Dec 2016 01:07:16 GMT

```
[
  {
    "bytes": 21122,
    "content_type": "text/html",
    "hash": "89022f6ffeed5ec72f32114f21301538",
    "last_modified": "2016-11-21T07:51:36.707180",
    "name": "java_docs/constant-values.html"
  },
  {
    "bytes": 5633,
    "content_type": "text/html",
    "hash": "98d25ffba777f1ff09eb81b9d5d4e214",
    "last_modified": "2016-11-21T07:51:36.770920",
    "name": "java_docs/deprecated-list.html"
  },
  {
    "bytes": 9198,
    "content_type": "text/html",
    "hash": "8ba11b28dc27d52d50803da4915bbd1d",
    "last_modified": "2016-11-21T07:51:36.701200",
    "name": "java_docs/help-doc.html"
  },
  {
```

```

        "bytes": 1454,
        "content_type": "text/html",
        "hash": "17111a7c72b5a029265fc3695e005ed7",
        "last_modified": "2016-11-21T07:51:36.701170",
        "name": "java_docs/index.html"
    },
    {
        "bytes": 10072,
        "content_type": "text/html",
        "hash": "8533c538282ce622e6aa40c514f25e5b",
        "last_modified": "2016-11-21T07:51:36.706890",
        "name": "java_docs/serialized-form.html"
    },
    {
        "bytes": 502,
        "content_type": "text/css",
        "hash": "3380eb278fdb37e63c45fbf7f542b47f",
        "last_modified": "2016-11-21T07:51:36.698970",
        "name": "java_docs/styleSheet.css"
    }
]

```

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

**Sample Request**

```

GET /v1/<account>/<container>?format=xml HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb

```

**Sample Response**

```

HTTP/1.1 200 OK
Content-Length: 1366
X-Container-Object-Count: 6
Accept-Ranges: bytes
X-Storage-Policy: Policy-0
X-Container-Bytes-Used: 47981
X-Timestamp: 1479714696.59461

```

```
Content-Type: application/xml; charset=utf-8
X-Trans-Id: txa5d051427ad843f2ba276-005840c994
Date: Fri, 02 Dec 2016 01:08:36 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<container name="con01"> <object> <name>java_docs/constant-
values.html</name> <hash>89022f6ffeed5ec72f32114f21301538</hash> <bytes>21122</byt
es> <content_type>text/html</content_type> <last_modified>2016-11-
21T07:51:36.707180</last_modified> </object> <object> <name>java_docs/deprecated-
list.html</name> <hash>98d25ffba777f1ff09eb81b9d5d4e214</hash> <bytes>5633</bytes>
<content_type>text/html</content_type> <last_modified>2016-11-
21T07:51:36.770920</last_modified> </object> <object> <name>java_docs/help-
doc.html</name> <hash>8ba11b28dc27d52d50803da4915bbd1d</hash> <bytes>9198</byte
s> <content_type>text/html</content_type> <last_modified>2016-11-
21T07:51:36.701200</last_modified> </object> <object> <name>java_docs/index.html</name
> <hash>17111a7c72b5a029265fc3695e005ed7</hash> <bytes>1454</bytes> <content_type
>text/html</content_type> <last_modified>2016-11-
21T07:51:36.701170</last_modified> </object> <object> <name>java_docs/serialized-
form.html</name> <hash>8533c538282ce622e6aa40c514f25e5b</hash> <bytes>10072</byt
es> <content_type>text/html</content_type> <last_modified>2016-11-
21T07:51:36.706890</last_modified> </object> <object> <name>java_docs/stylesheet.css</na
me> <hash>3380eb278fdb37e63c45fbf7f542b47f</hash> <bytes>502</bytes> <content_type
>text/css</content_type> <last_modified>2016-11-
21T07:51:36.698970</last_modified> </object> </container>
```

- **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 : ssproxy2.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
X-Storage-Policy: Policy-0
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/cats/persian.jpg
photos/animals/cats/siamese.jpg
photos/animals/dogs/poodle.jpg
photos/animals/dogs/terrier.jpg
photos/me.jpg
photos/plants/fern.jpg
```

```
photos/plants/rose.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 : ssproxy2.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 : ssproxy2.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. Container 생성 시에는 두 종류의 storage policy를 선택 할 수 있으며, storage policy를 입력하지 않으면, default policy 인 3copy로 설정된다. storage policy 종류는 standard(storage policy: 3copy와 economy(storage policy: ec)로 두 가지 타입을 선택적으로 사용 가능하다. 두 policy의 차이는 object 복제 방식으로, 자세한 내용은 포탈 소개 내용을 참고한다.

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
X-Storage-Policy	storage policy 종류로 economy 이용 시 ec 를 입력	String	No

**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(standard: default)**

```
PUT /v1/<account>/<container> HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : 5f6d351c490b44b8b60b015e744a435a
```

**Sample Response**

```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txf828c1ea8c024ccb9ef3a-005840d56c
Date: Fri, 02 Dec 2016 01:59:08 GMT
```

- **Example**

**Sample Request(economy)**

```
PUT /v1/<account>/<container> HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : 5f6d351c490b44b8b60b015e744a435a
x-storage-policy: ec
```

**Sample Response**

```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx72a9ee5b3e41433daa6cd-005840d6f9
Date: Fri, 02 Dec 2016 02:05:46 GMT
```

#### 4.4 DELETE Storage container

- **API 설명**

**container**를 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**

```
DELETE /v1/<account>/<container> HTTP/1.1
Host : ssproxy2.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-Length: 0
Content-Type: text/plain; charset=UTF-8
```

#### 4.5 POST Storage container : 사용자 metadata

- **API 설명**

**container metadata**를 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 : ssproxy2.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
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx53839635babf464f831af-005840d825
Date: Fri, 02 Dec 2016 02:10:45 GMT
```

- **Example2**

Example1 에서 추가한 container metadata 를 삭제하는 예이다.

#### Sample Request

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy2.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
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txb97d4e1aca0348488f66b-005840d85c
Date: Fri, 02 Dec 2016 02:11:40 GMT
```

## 4.6 POST Storage container : ACL

- API 설명

**container의 ACL을 설정한다.**

KT ucloud storage 2.0 service는 container별로 접근제어에 대한 설정이 가능하다. http post를 사용하여 사용자 metadata를 정의하는 방식과 동일하나, 특정 헤더 key를 사용해야한다. 현재 ACL은 전체 공개만 제공하고 있으며, header key는 'x-container-read'키를 사용해야만 한다.

자세한 내용은 아래 example을 참조

	example	description
referrer 형식	.r.*	임의의 host 접근 허용
	.r:*,.rlistings	임의의 host에 대해 container안에 있는 object 리스트 조회 허용

- Request Syntax

```
POST/<api version>/<account>/<container> HTTP/1.1
Host : 스토리지 서비스 서버
X-Auth-Token : 인증 토큰
x-container-read: [item[,item...]]
```

### Required Request Headers

Header Name	Description	Type	Required
X-Auth-Token	사용자 인증 Token	String	Yes
X-Container-Read	ACL을 설정하기 위한 http header key	String	Yes
X-Remove-Container- Read	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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : 5f6d351c490b44b8b60b015e744a435a
x-container-read: .r:*
```

### Sample Response

```
HTTP/1.1 204 No Content
```

```
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx008a6fea50c44f51a8dfd-005840ff6b
Date: Fri, 02 Dec 2016 04:58:19 GMT
```

## 4.7 POST Storage container static website

- **API 설명**

**container**를 **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 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 container 내 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	num	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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token: 5f6d351c490b44b8b60b015e744a435a
X-Container-Meta-Web-Index: index.html
```

**Static website error suffix 설정**

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token: 5f6d351c490b44b8b60b015e744a435a
X-Container-Meta-Web-Error: error.html
```

**HTML file listings 설정**

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token: 5f6d351c490b44b8b60b015e744a435a
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.

## 제 5 장 Storage object service

### 5.1 HEAD Storage object

- API 설명

**object metadata**를 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

**Sample Response**

```
HTTP/1.1 200 OK
Content-Length: 239
Last-Modified: Fri, 02 Dec 2016 05:49:09 GMT
Accept-Ranges: bytes
X-Object-Meta-Meat: Bacon
Etag: 9057dd33033c44505ee639a1d8fa7864
X-Object-Meta-Veggie: Bacon
X-Object-Meta-Fruit: Bacon
X-Timestamp: 1480657748.58168
Content-Type: application/octet-stream
X-Trans-Id: tx0231e52b24e84ce4bb986-0058410b58
Date: Fri, 02 Dec 2016 05:49:12 GMT
```

## 5.2 GET Storage object

- **API 설명**

**object**를 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
- o 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 : ssproxy2.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**

**object**를 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 스토리지 object API를 통해서 response header로부터 조회가 가능하다. 요청헤더에 Content-Length 및 Content-Type이 누락되어서는 안 된다.

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 : ssproxy2.ucloudbiz.olleh.com
```

```
X-Auth-Token : 5f6d351c490b44b8b60b015e744a435a
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 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.(데이터의 전체 크기를 알기 전에 동적으로 생성한 데이터를 전송할 수 있다.)

- **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

## 5.5 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 다시 해야만 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 0으로 지정해야만 한다.

- **Request Syntax**

```
PUT /<api version>/<account>/<container>/<destobject> HTTP/1.1
Host : ssproxy2.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

## 5.6 POST Storage object : 사용자 metadata

- API 설명

**object metadata**를 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 container 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. 만일 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 : ssproxy2.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: Fri, 02 Dec 2016 06:43:59 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

### Sample Response

```
HTTP/1.1 202 Accepted
Date: Fri, 02 Dec 2016 06:43:59 GMT
Content-Length: 0
Content-Type: text/plain; charset=UTF-8
```

## 5.7 DELETE Storage object

- **API 설명**

**object**를 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 : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

#### Sample Response

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txb0e7c2cd3d8f45c1a9307-005841193b
Date: Fri, 02 Dec 2016 06:48:27 GMT
```

## 제 6 장 Dynamic Large Objects(DLO)

- **API 설명**

**5GB 크기를 넘는 object 를 업로드한다.**

KT ucloud storage 2.0 service 에서 지원하는 최대 업로드 파일 크기는 5GB 이다. 따라서, 5GB 를 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. API 에서 직접적인 분할을 지원하지는 않는다.

(2) 작은 크기로 분할된 각 segment 를 업로드 한다. 개별 segment 를 업로드 할 경우

우, 반드시 동일한 container 에 업로드 해야 하고, object 이름에 동일한 prefix 를 가져야 한다. 또한, 각 segment 의 이름은 향후 다운로드 시 연결시키기 위해 순서적으로 정렬이 가능해야 한다.

개별 segment 업로드는 기존 업로드 API 를 활용하면 된다.

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

자세한 사용 예는 example 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 jls.data 파일이 있다. jls.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/jls.data/1.data HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

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

**두번째 segment 업로드**

```
PUT /v1/<account>/test_segments/jls.data/2.data HTTP/1.1
```

```
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

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

### 세번째 segment 업로드

```
PUT /v1/<account>/test_segments/jls.data/3.data HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

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

### manifest file 업로드

```
PUT /v1/<account>/test/jls.data HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Object-Manifest: test_segments/jls.data/
Content-Length: 0
```

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

### jls.data 다운로드

```
GET /v1/<account>/test/jls.data HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

### container ACL 추가

container에 전체공개를 적용했을 경우, 아래와 같이 test\_segments container 의 'x-container-read' header key도 설정 해 주어야 한다.

```
POST /v1/<account>/<container> HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : AUTH_tk5f6d351c490b44b8b60b015e744a435a
```



```
x-container-read: .r:* ,rlistings
```

## 제 7 장 Static Large Objects(SLO)

DLO(Dynamic Large Objects)와 비슷한 기능으로 사용자가 많은 object들을 동시에 업로드 하고, 하나의 object로 다운 받을 수 있도록 한다. 그러나 DLO와는 다르게 사용자가 object 의 segments들을 일일이 정의 해야 한다.

- **API 설명**

- **5GB 크기를 넘는 object 를 업로드한다.**

KT ucloud storage service 에서 지원하는 최대 업로드 파일 크기는 5GB 이다. 따라서, 5GB 를 넘는 파일의 업로드 및 다운로드를 다루기 위해서 몇 가지 절차를 수행 하면 된다.

- **업로드 방법 및 절차**

- (1) 5GB 크기를 넘는 파일을 먼저 5GB 이하의 작은 크기로 분할한다. 이때, 분할의 크기 및 수량에는 제한이 없다. API 에서 직접적인 분할을 지원하지는 않는다.
- (2) 작은 크기로 분할된 각 segment 를 업로드 한다. 개별 segment 를 업로드 할 경우, DLO 와는 다르게 container 의 위치와 object 의 이름에 제약이 없다. 또한, 각 object 를 특정 순서대로 업로드 할 필요도 없다.  
개별 segment 업로드는 기존 업로드 API 를 활용하면 된다.
- (3) 마지막으로 manifest 파일을 업로드 해야 한다. manifest 파일을 업로드 할 때에는 아래의 query parameter 와 함께 업로드 해야 하며,

```
?multipart-manifest=put
```

요청의 body 부분에는 segment 들의 목록을 JSON 형식으로 작성한다.

자세한 사용 예는 example 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 js.data 파일이 있다. js.data 파일을 3GB 파일 3개로(1.data, 2.data, 3.data) 분할 했다. 각 segment와 manifest 파일을 원하는 위치에 업로드 한다.

**첫번째 segment 업로드**

```
PUT /v1/<account>/tes01/segment HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

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

### 두번째 segment 업로드

```
PUT /v1/<account>/tes02/list/something HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

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

### 세번째 segment 업로드

```
PUT /v1/<account>/jls/data HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Content-Length: 3221225472
```

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

### manifest file 업로드

```
PUT /v1/<account>/test03/jls.data?multipart-manifest=put HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

```
[
  {
    "path": "test01/segment",
    "etag": "4322d422199eaccfbd36feb3da955b16",
    "size_bytes": 3145728000
  },
  {
    "path": "test02/list/something",
    "etag": "4322d422199eaccfbd36feb3da955b16",
    "size_bytes": 3145728000
  },
  {
    "path": "jls/data",
```

```
    "etag": "99bd40956a312dcfd4e76dd16725d640",
    "size_bytes": 2924544000
  }
]
```

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

### jls.data 다운로드

```
GET /v1/<account>/test/jls.data HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
```

jls.data 파일의 메타데이터를 보면 'X-Static-Large-Object: True' 설정 되어 있는 것을 확인할 수 있다

뒤에 query parameter를 추가하면 원래의 manifest json body를 확인할 수 있다.

```
?multipart-manifest=get
```

### jls.data 삭제

jls.data에 대해 query parameter를 추가하면 manifest 파일 body에 참조 된 각 segment 파일들이 삭제 된 후에, jls.data 파일이 삭제된다.

```
?multipart-manifest=delete
```

## 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
<a href="https://ssproxy2.ucloudbiz.olleh.com/v1/account/container/object?temp_url_sig=sig&amp;temp_url_expires=expires">https://ssproxy2.ucloudbiz.olleh.com/v1/account/container/object?temp_url_sig=sig &amp;temp_url_expires=expires</a>

### 8.4 TempURL 요청

- GET Request Syntax

GET /v1/ <b>account/container/object</b> ?temp_url_sig= <b>sig</b> &temp_url_expires= <b>expires</b>
HTTP/1.1
Host : https://ssproxy2.ucloudbiz.olleh.com

- PUT Request Syntax

GET /v1/ <b>account/container/object</b> ?temp_url_sig= <b>sig</b> &temp_url_expires= <b>expires</b>
HTTP/1.1
Host : https://ssproxy2.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/ <b>account/container/object</b> ?temp_url_sig= <b>sig</b> &temp_url_expires= <b>expires</b>
HTTP/1.1
Host : https://ssproxy2.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://ssproxy2.ucloudbiz.olleh.com/v1/AUTH\\_account/container/object\\_prefix](https://ssproxy2.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://ssproxy2.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
PUT	Object 업로드 시에 Expiring Objects 설정
POST	기 업로드된 Object에 Expiring Objects 설정

- 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 : ssproxy2.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 : ssproxy2.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 : ssproxy2.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 : ssproxy2.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 : ssproxy2.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
PUT	Container 생성 시에 CORS 설정

POST	존재하는 Container에 CORS 설정
------	-------------------------

CORS 메타데이터를 추가한다.

**Sample Request**

```
PUT /v1/account/container HTTP/1.1
Host : ssproxy2.ucloudbiz.olleh.com
X-Auth-Token : eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
X-Container-Meta-Access-Control-Allow-Origin: http://kt-dev-test.net (원본 웹서버)
```

## 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 : ssproxy2.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 : ssproxy2.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
```