WebDAV

～オープンデータ時代の標準API～

2007/01/15 ウノウ株式会社 駒形真幸 <komagata@unoh.net>

WebDAVとは？

- **D**istributed **A**uthoring and **V**ersioning protocol for the WWW
- WWW上で編集とバージョン管理が出来るプロトコル
- 1998年にRFC2291で提唱されたHTTP1.1の拡張仕様

主な実装

- サーバ
  Apache mod_dav
  IIS

- クライアント
  WindowsのExplorer
  Mac OS XのFinder
  LinuxのNautilus
  Dreamweaver

WebDAVの基本要素

- リソース
  WebDAVで扱う対象
  （ファイルシステムで言えばファイルやディレクトリ）
- コレクション
  リソースの集まり（ファイルシステムで言えばディレクトリ）
- プロパティ
  リソースの属性（名前と値のペア）
- ロック
  同時編集を回避する仕組み（共有ロック方式）

HTTP1.1からの拡張点

- メソッドの追加

メソッド	機能
PROPFIND	プロパティの取得
PROPPATCH	プロパティの変更
MKCOL	コレクションの作成
COPY	コレクションを含むリソースおよびプロパティの複製
MOVE	コレクションを含むリソースの移動
LOCK	リソースのロック
UNLOCK	リソースのロック解除

HTTP1.1からの拡張点（2）

- ステータスコードの追加

値	ステータス	意味
102	Processing	リクエストは受け付けたが、まだ処理が終わっていない
207	Multi-Status	複数のステータスを持つ
422	Unprocessible Entity	リクエストの書式は正しいが、その内容が間違っている
423	Locked	リソースはロックされている
424	Failed Dependency	あるリクエストに関連したリクエストが失敗したため、依存関係が保てない
507	Insufficient Storage	記憶領域が不足している

WebDAVのリクエスト

```
PROPFIND /DAV/setup01.jpg HTTP/1.1
Host: localhost
Content-type: text/xml; charset="utf-8"
Depth: 0


<?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop xmlns:R="http://localhost/boxschema/">
    <R:bigbox/>
      <R:author/>
      <R:DingALing/>
    <R:Random/>
  </D:prop>
</D:propfind>
```

WebDAVのレスポンス

```
HTTP/1.1 207 Multi-Status
Date: Thu, 18 Jan 2007 02:55:37 GMT
Server: Apache/2.2.3 (Debian) DAV/2 PHP/4.4.4-8
Content-Length: 1245
Connection: close
Content-Type: text/xml; charset="utf-8"


<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
<D:response xmlns:ns0="DAV:" xmlns:ns1="urn:schemas-microsoft-
com:" xmlns:lp1="DAV:" xmlns:lp
2="http://apache.org/dav/props/">
<D:href>/dav/a.jpg</D:href>
 （略）
```

枯れた地味な技術

- 新しいファイルシステムはもういいよ･･･
- nfsで良いんじゃ？
- 一昔前の技術？
- 今日びWebDAVて･･･

なぜ今、WebDAVか

- 良く考えると、WebDAVは
  ファイルシステムじゃなくてWeb APIの仕様

- もしかしてスゴイ有望なんじゃ？

なぜ今、WebDAVか（2）

- 様々なデバイスでネットにアクセス
- オープンなAPIよりオープンなデータ
- CGMの流行でネット上のWRITEが増えた

なぜ今、WebDAVか（3）

- 様々なデバイスでネットにアクセス
  → HTTP1.1の拡張仕様
- オープンなAPIよりオープンなデータ
  → データ向けのXML語彙
- CGMの流行でネット上のWRITEが増えた
  → LOCKなどのWRITE向けの便利な仕様

WebDAV API

Web APIとしてのWebDAV

REST

- REpresentational State Transfer

URIで識別されるリソースの状態の表現をGET, PUTなどの幾つかの動詞によってやりとりするアーキテクチャースタイル

HTTPプロトコルの主要著者であるRoy Fieldingが提唱した。WWWと非常に相性が良い。（WWW自体RESTの一実装であると言える）

RESTなシステム

- Google Web API
- A9.comのOpenSearch
- Blogger API, MetaBlog API
- Atom Publishing Protocol(AtomPP)
- WebDAV！？

WebDAV vs AtomPP

- Web APIの仕様っていったらAtomPPじゃないの？

→ WebDAVは1999年にRFCになっている。
対してAtomPPは現在IETFで議論中。
現状、実装も圧倒的にWebDAVが多い。

WebDAV vs AtomPP（2）

- RESTfulなAtomPPの方がWWWと相性が良いんじゃないの？

Roy FieldingのREST原則

- ステートレスなプロトコル（HTTP)
- 全てのリソースに適用可能な操作のセット（GET, POST, PUT, DELETEなど）
- リソースを一意に識別できる汎用的な構文（URI）
- 情報と状態遷移の両方を扱うことができる「ハイパーメディアの使用」（HTMLやXML）

WebDAVはRESTful

- HTTP 1.1の拡張
- PROPFIND, COPYなど全てのリソースに適用できる汎用のメソッドを持っている
- URIをベースとしている
- XMLを使ってやり取りしている

WebDAV vs AtomPP（3）

- 本当に普及してるの？

→ Windows, Mac OS X, Linuxで
標準でサポートしている。
実装も圧倒的にWebDAVの方が多い。

WebDAVの利点

- 実は一番普及しているWeb API
- ファイルシステムを扱うのに十分なメソッドと表現力
- 詳細なわりに抽象度が高い仕様
（ファイルシステム自体の抽象度が高いので必然的にこうなった）

PHPよるサーバ実装

対応するメソッドを実装するだけで良い

```
require_once 'HTTP/WebDAV/Server.php';
class HTTP_WebDAV_Server_Photozou extends
HTTP_WebDAV_Server {
  function GET(&$options) {
    $options['mimetype'] = 'image/jpeg';
    $options['mtime'] = xxxxxx;
    $options['stream'] = fopen($options['path'], 'r');
      return true;
  }
  function PROPFIND(&$options, &$files) { (略) }
}
$server = new HTTP_WebDAV_Server_Photozou();
$server->ServeRequest();
```

Javascriptによるクライアント実装

XHR(Xml Http Request)でWebDAVの拡張メソッドを呼ぶことが出来る

```
var XHR =  function() {
  return window.XMLHttpRequest ?
    new XMLHttpRequest() : new ActiveXObject("Microsoft.XMLHTTP");
}
XHR.open('PROPFIND', 'http://foo.com/pathto, true);
headers = headers || {};
headers['Depth'] = headers['Depth']>1 ? 1 : headers['Depth'];
headers['Content-Type'] = headers['Content-Type'] || 'text/xml';
for (h in headers) {
  XHR.setRequestHeader(h, headers[h]);
}
XHR.send('<?xml version="1.0" ?><propfind
xmlns="DAV:"><allprop/></propfind>');
```

認証

RESTなので既存の様々な認証方法が使えます。

- Basic認証

```
PROPFIND /pathto HTTP/1.1
Depth: 1
Authorization: Basic a28tYWdhdGE6dGFrYXNha2k=
```

- Digest認証

```
PROPFIND /pathto HTTP/1.1
Depth: 1
WWW-Authenticate: Digest realm="Secret Zone",
nonce="RMH1usDrAwA=6dc290ea3304de42a7347e0a94089ff5912ce0de", algorithm=MD5, qop="auth"
```

認証（2）

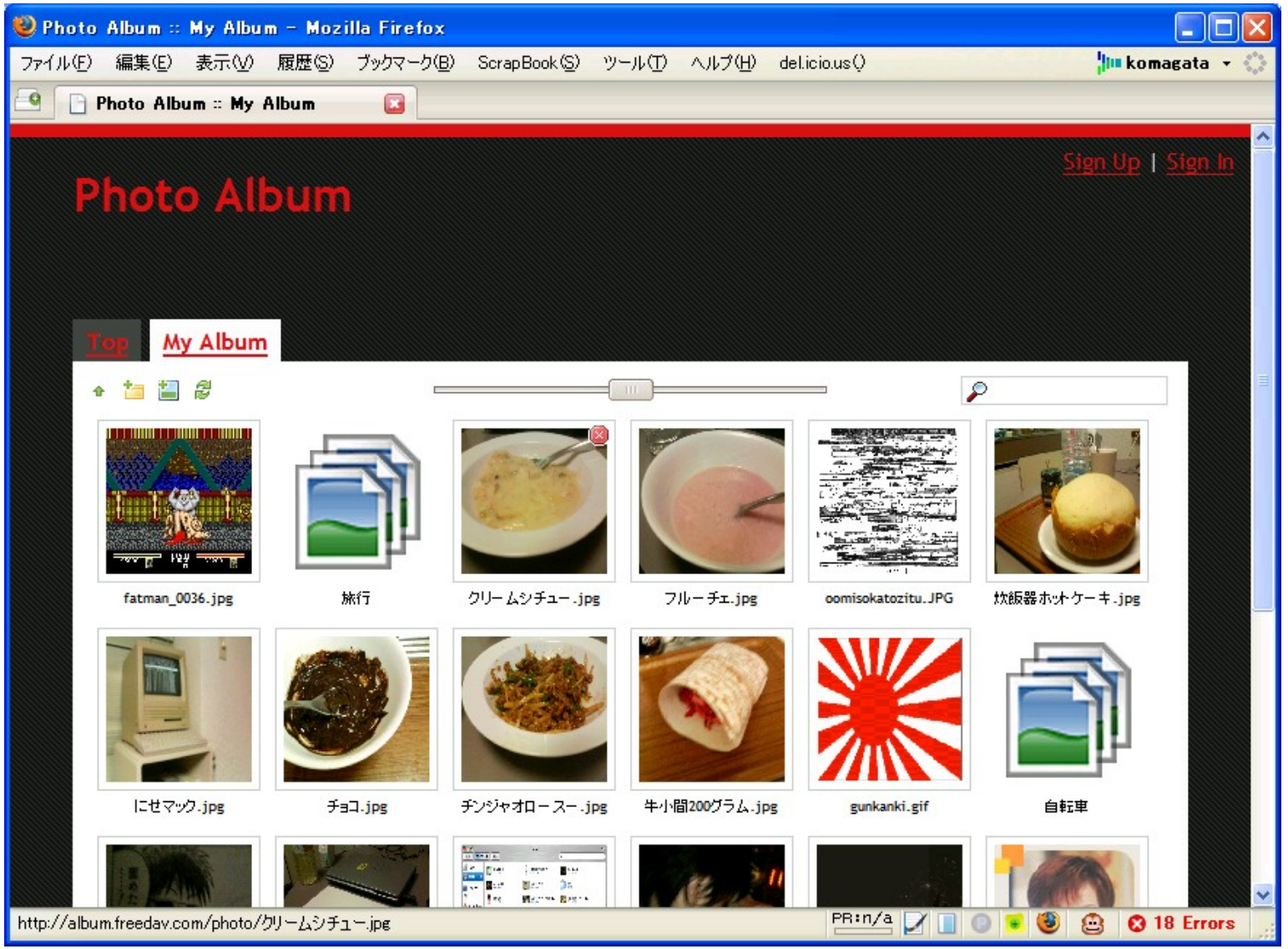
- WSSE認証

```
PROPFIND /pathto HTTP/1.1
Depth: 1
X-WSSE: UsernameToken Username="komagata",
PasswordDigest="ZCNaK2jrXr4+zsCaYK/YLUxImZU=",
Nonce="Uh95NQIviNpJQR1MmML+zq6pFxE=", Created="2005-01-
18T03:20:15Z"
```

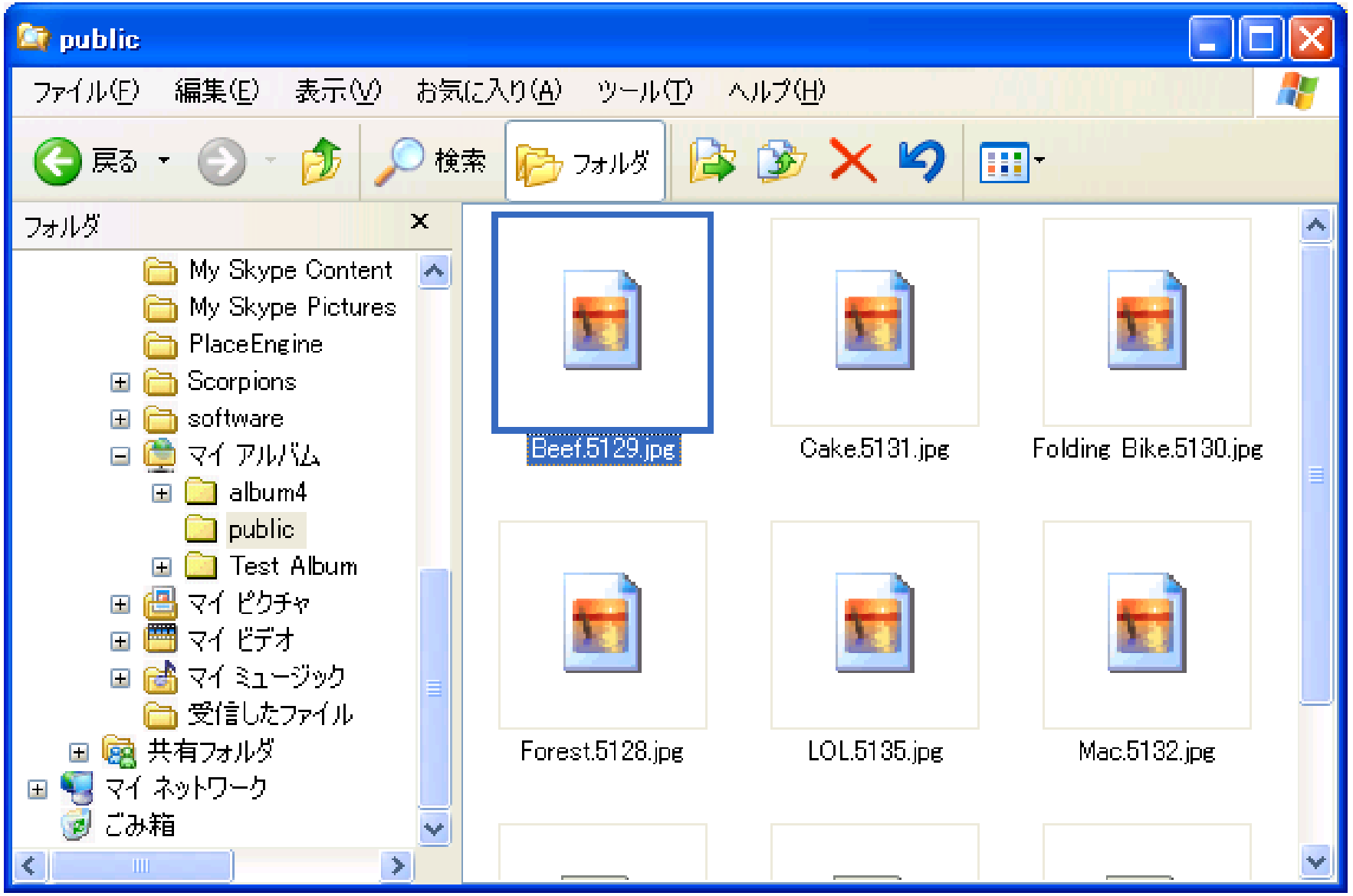
WebDAV API応用例

- JavaScriptの画像ブラウザ
- フォト蔵をWindowsにマウント
- JS画像ブラウザ at フォト蔵
- Flashの画像ブラウザ

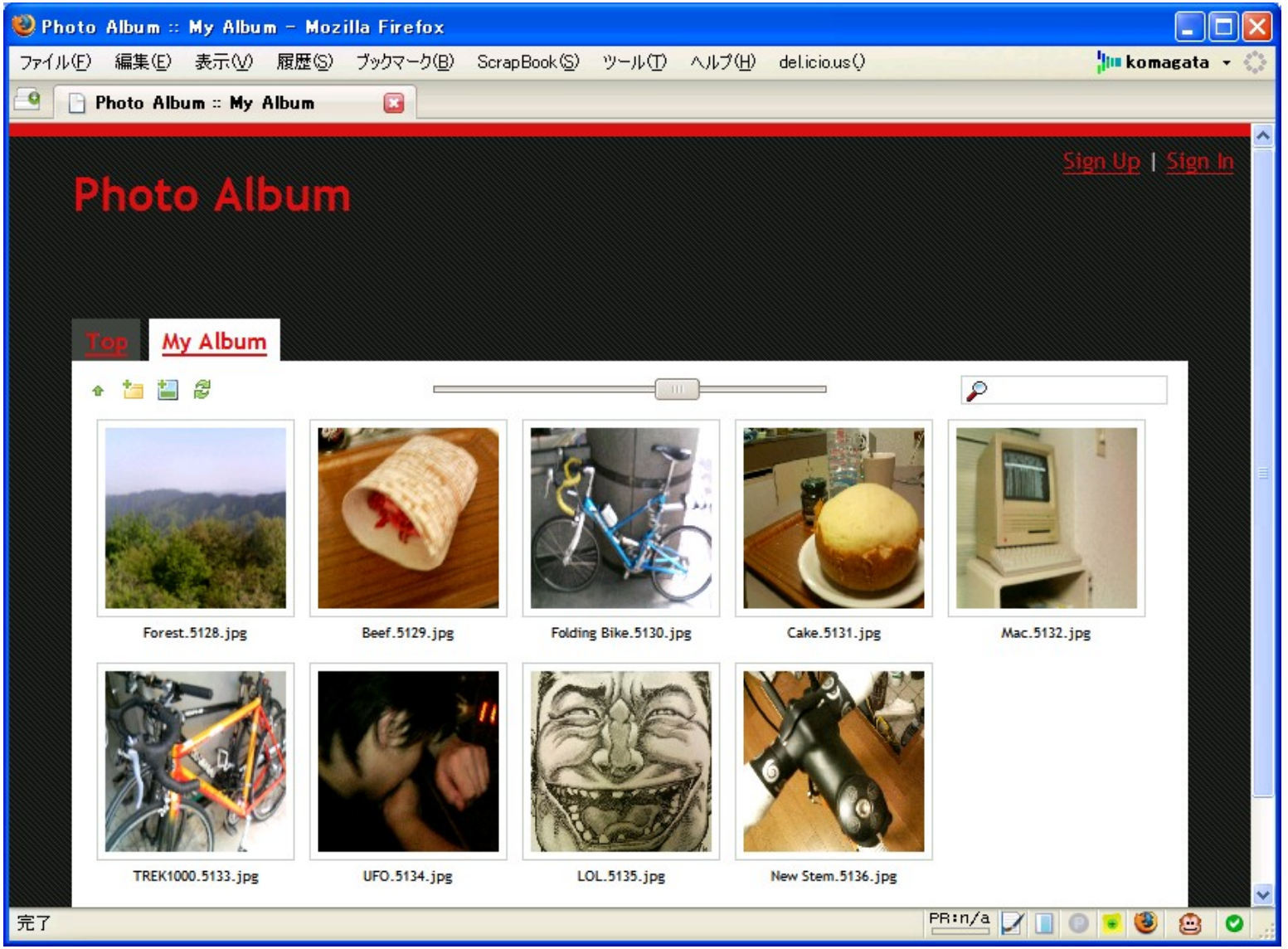
JavaScriptの画像ブラウザ



フォト蔵をWindowsにマウント



JS画像ブラウザ at フォト蔵



JS画像ブラウザ at フォト蔵

Flashの画像ブラウザ

まとめ

WebDAVは有望なWeb API