

XML-RPC インターフェース

目的

- ・FSWiki において XML-RPC によるデータのやりとりを可能にする
- ・上とは全然関係ないのですが、Web 上のテキストをクリック一つ（から二つ）で Wiki に書き込めるようなツールの提供

ファイル構成

主要ファイル

wiki_xmlrpc.cgi

サーバ CGI。wiki.cgi と同じディレクトリに設置します。

lib/WikiAPI.pm

API 用モジュール。

lib/WikiAPI/WikiMod.pm

WikiAPI.pm と Wiki.pm を取り持つアダプタみたいなもの

lib/WikiAPI/LinkParser.pm

linkList（後述）用のパーサ

lib/WikiAPI/Introspection.pm

イントロスペクション用ファイル

CPAN モジュール群

- ・SOAP
- ・XML
- ・XMLRPC

を利用しています。Perl5.00503、5.8、5.8 with mod_perl(Apache::Registry) で動作確認しています。lib 以下にいます。

おまけ

wikigate.cgi

WikiMod.pm を利用した簡易インターフェース(XML-RPC とは関係ありません)。wiki.cgi と同じディレクトリに置きます。

etc/wikitest.html

IE 用 bookmarklet。ブラウザ上で選択したテキストを直ちに FSWiki のページとして(追加)保存します。実験的なもので IE でしか動きません。wikigate.cgi を叩きます。

etc/xmlrpc_client.cgi

適当クライアント CGI。使う必要はありません。パラメータは wikigate.cgi に準じます。

設置方法

ファイルを展開して、稼働中の wiki ディレクトリに一式入れます。標準で 3.5.3 対応ですが、一行書き直すだけで 3.5.2 でも動作することを確認しています。

使用可能 API

下記以外にも一応インターフェースはありますが、FSWiki にはページのバージョンという概念がないと思うので、実装していません。また、ページのバージョンに関するデータは 1 を、ページの作者については空文字を返します。

```
array wiki.getRecentChanges( Date timestamp [,String farmname] )
```

timestamp(UTC) 以降変更されたページのリストを返します。WikiFarm に対応するため、第二引数 (wikifarm 名) をとれます。

戻り値: 構造 { name, lastModified, author, version } の配列リファレンス (name は utf8 で URL エンコードされています。author は常に空文字、version は常に 1)。

```
int wiki.getRPCVersionSupported()
```

戻り値: この API のバージョンを返します。

```
base64 wiki.getPage( String pagename )
```

ページのソースを返します。ページにコロン ':' が入っているとコロンより前部分が WikiFarm のパスになります。 ('farmname:pagename')

戻り値: ソーステキスト

```
base64 wiki.getPageHTML( String pagename )
```

HTML レンダリングされたページを返します。ページにコロン ':' が入っているとコロンより前部分が WikiFarm のパスになります ('farmname:pagename')。

戻り値: レンダリング済みテキスト

```
array wiki.getAllPages( [String farmname] )
```

全ページのリストを返します。WikiFarm に対応するため第二引数に Farm 名をとれます。

戻り値: 配列リファレンス (ページ名は utf8 で URL エンコードされています)。

```
struct wiki.getPageInfo( string pagename )
```

ページ情報を返します。

戻り値: 構造 { name, lastModified, author, version } (name は utf8 で URL エンコードされています。author は常に空文字、version は常に 1)。

```
array wiki.listLinks( string pagename )
```

与えられたページの全リンクを返します。

戻り値: 構造 { page, type, href } の配列リファレンス(page はリンク先の名前(utf8 で URL エンコード) か URL です。 type は external (外部) か local (内部) href はリンク先の HREF)

```
boolean wiki.putPage( String pagename, base64 content, struct attributes )
```

ページに内容を書き込みます。 pagename にコロン (:) を含めると前部分が WikiFarm 名になります。 pagename は utf8 で URL エンコードされている必要があります。現在の所、attributes で「下げ」指定ができます。 { minoredit => 0|1 }

また、attributes に id,pass を設定することで、ログインユーザとして書き込むことができます。

戻り値 : 成功時 1、それ以外は 0

wikigate.cgi 用のパラメータ

主に WikiAPI::WikiMod を利用するだけなので、XML-RPC とは関係ない代物です。

```
wikigate.cgi?method=addPage&page=Memo&sage=1&content=ほにやら
```

などとやりますと、[Memo](#) というページにタイムスタンプを更新することなくテキストを追加できます。で、これを利用して [bookmarklet](#) をつくったりとかしたら便利かなあと。一応 IE 用に一つテスト用ファイルが含まれています。

method

呼び出したい手続き名。API と同じ名前を持ちますが、大文字小文字は関係有りません。また、addpage という値もとることができ、これは putpage と違い、既存のページに追加をします。自動で追加データの前に改行されます。

page

ページ名。ページ名に ':' をつけることで、コロンの前の部分が WikiFarm 名になります。

farm

WikiFarm 名。このパラメータは recentchanges、getallpages に対してのみ有効です。

day

recentchanges において指定する過去の日数。day=3 なら 3 日前からの更新ページの一覧が表示されます。

sage

真値を設定することで、putpage、addpage におけるタイムスタンプを更新しなくなります。

add

真値を設定することで、putpage の場合でも追加書き込みにする。

content

putpage、addpage で書き込むデータ。

id

ログインユーザしか書き込めない設定の場合に putPage、addPage で使用するユーザ id。

pass

ログインユーザしか書き込めない設定の場合に putPage、addPage 時で使用するパスワード。

注意

- ・管理者にしか見られないページは、見られません。
- ・wiki_xmlrpc.cgi が利用している XML::Parser::Lite は公式には Perl5.6 以降対応となっているのですが、5.00503 でも動作することを確認しています（必ずしも動くわけではないかもしれませんが）。[参照](#)
- ・3.5.2 で利用する際は、WikiMod.pm の DESTROYD メソッドの一行をコメントアウトしてください。

```
sub DESTROY{ # この一行をコメントアウト
  # $_[0]->exit(); # before 3.5.2, this must be commented out
}
```

TODO

- ・bookmarklet の追加。