WikiLink` package

Version 0.2, April, 2014, Dror Bar-Natan and Scott Morrison

Introduction

Allows for direct manipulation of wiki pages from within mathematica.

Pensieve header: WikiLink` package - for direct manipulation of wiki pages from within mathematica.

Implementation

```
(*BeginPackage["WikiLink`"];*)

WikiConnection::usage = "WikiConnection[---] is a
    wiki connection, usually created with CreateWikiConnection.";

$WikiConnection::usage =
  "$WikiConnection stores the most recent wiki connection created
    with CreateWikiConnection. It is the default wiki connection for
    several other commands, if no other wiki connection is specified.";

CreateWikiConnection::usage =
  "CreateWikiConnection[URL, username, password] initialises a connection
    to a mediawiki server, returning a WikiConnection onject and
    resetting $WikiConnection. The username and password are optional.";

WikiGetPageText::usage =
  "WikiGetPageText[wikiconnection, title] returns the raw text of the
    specified page. WikiGetPageText[title] uses the default wiki
    connection stored at $WikiConnection. If the parameter \"title\" is
    a list, the result will be a list of the corresponsing raw texts.";

WikiGetPageTexts::usage =
  "WikiGetPageTexts[{title1, title2, ...}] returns a list of pairs of the form
    {{title1, text1}, {title2, text2}, ...}. The wiki connection may be
    specified using WikiGetPageTexts[wikiconnection, {title1, title2, ...}].";

WikiSetPageText::usage =
  "WikiSetPageText[title, text] overwrites the contents of the specificied
    page with the given text. WikiSetPageText[title, text, summary]
    overwrites the contents of the specificied page with the given
    text and notes summary in the change log. The wiki connection
    may be specified using WikiSetPageText[wikiconnection, title,
    text] or WikiSetPageText[wikiconnection, title, text, summary]";
```

```
WikiSetPageTexts::usage =
  "WikiSetPageTexts[{{title1, text1},{title2,text2},...}] efficiently sets
    multiple pages, by first checking which texts are already up to date.
    A `summary' field may be added to some {title, text} pairs (making
    them triples) or specified globally using WikiSetPageText[{{title1,
    text1},{title2,text2},...}, summary]. The wiki connection may
    be specified using WikiSetPageTexts[wikiconnection, ...].";

WikiUploadFile::usage =
  "WikiUploadFile[name, description] uploads the specified file
    to the wiki. The wiki connection may be specified
    using WikiUploadFile[wikiconnection, ...].";

(*Begin["`Private`"];*)
```

WikiConnection

```
WikiConnection[info___][param_] := param /. {info};
```

CreateWikiConnection

```
CreateWikiConnection::NotImplemented =
  "Anonymous connections are not yet implemented.";
CreateWikiConnection[url_String] :=
  Message[CreateWikiConnection::NotImplemented];
CreateWikiConnection[url_String, username_String, password_String] := Module[
  {apiurl, content1, cookies1, parsedcontent1,
   content2, cookies2, content3, cookies3, edittoken},
  apiurl = url <> "/api.php";
  {content1, cookies1} = URLFetch[apiurl, {"Content", "Cookies"},
    "Parameters" -> {
      "action" -> "login",
      "format" → "xml", "lgname" → username, "lgpassword" → password
     },
    "Method" → "POST", "StoreCookies" → False
   ];
parsedcontent1 = Cases[ImportString[content1, "XML"], XMLElement["login",
       {"result" → "NeedToken", parsedcontent___}, {}] :→ {parsedcontent},
     Infinity
    ] // First;
{content2, cookies2} = URLFetch[apiurl, {"Content", "Cookies"},
    "Parameters" -> {
      "action" → "login", "format" → "xml",
      "lgname" → username, "lgpassword" → password,
      "lgtoken" → ("token" /. parsedcontent1)
     },
    "Method" → "POST", "Cookies" → cookies1, "StoreCookies" → False
   ];
  {content3, cookies3} = URLFetch[apiurl, {"Content", "Cookies"},
    "Parameters" -> {"action" → "tokens", "format" → "xml"},
    "Method" → "POST", "Cookies" → cookies2, "StoreCookies" → False
   ];
  edittoken = Cases[ImportString[content3, "XML"],
     XMLElement["tokens", {"edittoken" → edittoken_}, {}] :→ edittoken,
     Infinity
    ] // First;
  $WikiConnection = WikiConnection["url" → url, "apiurl" → url <> "/api.php",
    "username" → username, "cookies" → cookies2, "edittoken" → edittoken]
 ]
```

WikiGetPageText, WikiGetPageTexts

```
WikiGetPageText[title_] := WikiGetPageText[$WikiConnection, title];
WikiGetPageText[wc_WikiConnection, title_String] := URLFetch[
      wc@"apiurl",
      "Parameters" -> {"action" → "query", "prop" → "revisions",
        "titles" → title, "rvprop" → "content", "format" → "xml"},
      "Method" → "GET", "StoreCookies" → False
     ] //
     ImportString[#, "XML"] & //
    Cases[#, XMLElement["rev", _, {content_String}] :> content, Infinity] & //
   First;
WikiGetPageText[wc_WikiConnection, titles_List] :=
  WikiGetPageText[wc, #] & /@ titles;

WikiGetPageTexts[titles_List] := WikiGetPageTexts[$WikiConnection, titles];
WikiGetPageTexts[wc_WikiConnection, titles_List] := Thread[{
   titles,
   WikiGetPageText[wc, titles]
  }]
```

WikiSetPageText, WikiSetPageTexts

```
WikiSetPageText[title_String, etc___] :=
  WikiSetPageText[$WikiConnection, title, etc];
WikiSetPageText[wc_WikiConnection, title_String, text_String] :=
  WikiSetPageText[wc, title, text, ""];
WikiSetPageText[wc_WikiConnection, title_String, text_String, summary_String] :=
 URLFetch[wc@"apiurl",
  "Parameters" -> {"action" → "edit", "title" → title, "summary" → summary,
    "text" → text, "format" → "xml", "token" → wc@"edittoken"},
  "Method" → "POST", "Cookies" → wc@"cookies", "StoreCookies" → False
 ]

WikiSetPageTexts[l_List, etc___] := WikiSetPageTexts[$WikiConnection, l, etc];
WikiSetPageTexts[wc_WikiConnection, l : {{_String, _String} ...},
   summary_String] := WikiSetPageTexts[wc, Append[#, summary] & /@ l];
WikiSetPageTexts[wc_WikiConnection, l : {{_, __} ...}] := Module[{oldtexts},
  oldtexts = Last /@ WikiGetPageTexts[First /@ l];
  DeleteCases[Table[
    If[oldtexts[[k]] === l[[k, 2]], 0,
     WikiSetPageText @@ Prepend[l[[k]], wc]; k
    ],
    {k, Length@l}
   ], 0]
 ]

(*End[];*)
```

```
(*EndPackage[];*)
```