```
SetDirectory["C:\\drorbn\\AcademicPensieve\\2014-04\\MathematicaWikiLink"];
<< WikiLink2.m;
urls = {"http://drorbn.net/", "http://katlas.org/w/"};
url = urls[[2]];
username = "DrorsRobot";
Clear[password];
password := password = InputString["Password for " <> username <> ": "]
```

```
? WikiConnection
```

WikiConnection[---] is a wiki connection, usually created with CreateWikiConnection.

```
? $WikiConnection
```

$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
```

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

```
CreateWikiConnection[url]
```

CreateWikiConnection::NotImplemented : Anonymous connections are not yet implemented.

```
CreateWikiConnection[url, username, password]
```

```
WikiConnection[url → http://katlas.org/w/,
 apiurl → http://katlas.org/w//api.php, username → DrorsRobot,
 cookies → {{Domain → .wolfram.com, Path → /, Secure → FALSE, Expires →
     Wed 4 Feb 2026 20:55:38, Name → WR_SID, Value → 806444f3c6c56b7af4aa4a3},
   {Domain → #HttpOnly_katlas.org, MachineAccess → FALSE, Path → /,
    Secure → FALSE, Expires → Thu 1 Jan 1970 00:00:00,
    Name → wikidb_mw__session, Value → 8ffb69e8764a13ebca1413a85c070af0},
   {Domain → #HttpOnly_katlas.org, MachineAccess → FALSE, Path → /, Secure → FALSE,
    Expires → Fri 5 Aug 2016 21:13:36, Name → wikidb_mw_UserID, Value → 5},
   {Domain → #HttpOnly_katlas.org, MachineAccess → FALSE, Path → /,
    Secure → FALSE, Expires → Fri 5 Aug 2016 21:13:36, Name → wikidb_mw_UserName,
    Value → DrorsRobot}, {Domain → #HttpOnly_katlas.org, MachineAccess → FALSE,
    Path → /, Secure → FALSE, Expires → Fri 5 Aug 2016 21:13:36,
    Name → wikidb_mw_Token, Value → 8fdcea5b2577efb5d56879fe20e70a0a}},
 edittoken → 1ebf41e07a1a3f585e57fc9b82d82c65+\]
```

```
? WikiGetPageText
```

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.

```
WikiGetPageText["The_Mathematica_Package_KnotTheory`"]
```

```
{{Manual TOC Sidebar}}

This manual describes the [http://www.wolfram.com/ Mathematica]
  package KnotTheory`, the main tool used to produce The Knot Atlas.

[[Printable Manual|Printable version]]

[[Image:KnotTheory 240.gif|frame|center|[[Setup|Download / Setup]]]]
```

```
WikiGetPageText[
 {"The_Mathematica_Package_KnotTheory`", "Template:Manual TOC Sidebar"}]
```

```
{{{Manual TOC Sidebar}}

This manual describes the [http://www.wolfram.com/ Mathematica]
   package KnotTheory`, the main tool used to produce The Knot Atlas.

[[Printable Manual|Printable version]]

[[Image:KnotTheory 240.gif|frame|center|[[Setup|Download / Setup]]]],
 {{If not pagename transclude|Printable Manual|Manual_TOC_Sidebar2}}}
```

```
? WikiGetPageTexts
```

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, ...}].

```
WikiGetPageTexts[
 {"The_Mathematica_Package_KnotTheory`", "Template:Manual TOC Sidebar"}]
```

```
{{The_Mathematica_Package_KnotTheory`, {{Manual TOC Sidebar}}

This manual describes the [http://www.wolfram.com/ Mathematica]
    package KnotTheory`, the main tool used to produce The Knot Atlas.

[[Printable Manual|Printable version]]

[[Image:KnotTheory 240.gif|frame|center|[[Setup|Download / Setup]]]]},
 {Template:Manual TOC Sidebar,
  {{If not pagename transclude|Printable Manual|Manual_TOC_Sidebar2}}}}
```

```
? WikiSetPageText
```

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]

```
{
 DateString[],
 WikiSetPageText["sandbox",
  "Sandbox by WikiLink2 on " <> DateString[], "Just testing " <> DateString[]]
}
```

```
{Sun 6 Jul 2014 16:19:13,
 <?xml version="1.0"?><api><edit result="Success" pageid="1650053"
   title="Sandbox" contentmodel="wikitext" oldrevid="1721013"
   newrevid="1721014" newtimestamp="2014-07-06T20:19:14Z" /></api>}
```

```
? WikiSetPageTexts
```

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, ...].

```
{
 DateString[],
 WikiSetPageTexts[{
    {"sandbox", "Sandbox by WikiLink2 on " <> DateString[]},
    {"sandbox1", "Sandbox1 by WikiLink2 on " <> ToString[$SessionID]}
  }, "Just testing " <> DateString[]]
}
```

```
{Sun 6 Jul 2014 16:56:33, {1}}
```

WikiUploadFile - in progress

```
? WikiUploadFile
```

WikiUploadFile[name, description] uploads the specified file to
the wiki. The wiki connection may be specified using WikiUploadFile[wikiconnection, ...].

```
BinaryReadList["SandboxDate.png"] // Short
```

```
{137, 80, 78, 71, 13, 10, 26, 10, 0, 0, 0, 13, 73,
 <<857>>, 251, 0, 0, 0, 0, 73, 69, 78, 68, 174, 66, 96, 130}
```

```
WikiUploadFile[name_String, description_String] :=
  WikiUploadFile[$WikiConnection, name, description];
WikiUploadFile[wc_WikiConnection, name_String, description_String] :=
  URLFetch[wc@"apiurl",
   "Parameters" -> {"action" → "upload", "filename" → name,
     "text" → description, "format" → "xml", "token" → wc@"edittoken"},
   "MultipartData" → {{"file", "application/octet-stream", BinaryReadList[name]}},
   "Method" → "POST", "Cookies" → wc@"cookies", "StoreCookies" → False
  ];
{ds = DateString[],
 name = Export["SandboxDate.png", ds],
 WikiUploadFile[name, "SandboxDate.png by WikiLink2 on " <> ds]
}
```

```
f @@@ {1 → 2, 2 → 3}
```

```
{f[1, 2], f[2, 3]}
```

```
WikiUploadFile[name_String, description_String] :=
  WikiUploadFile[$WikiConnection, name, description];
WikiUploadFile[wc_WikiConnection, name_String, description_String] :=
  URLFetch[wc@"apiurl",
   "MultipartData" → Append[
     {#1, "text/plain", ToCharacterCode@#2} & @@@ {
       "action" → "upload", "filename" → name,
       "text" → description, "format" → "xml", "token" → wc@"edittoken"
      },
     {"file", "application/octet-stream", BinaryReadList[name]}
    ],
   "Method" → "POST", "Cookies" → wc@"cookies", "StoreCookies" → False
  ];
{ds = DateString[],
 name = Export["SandboxDate.png", ds],
 WikiUploadFile[name, "SandboxDate.png by WikiLink2 on " <> ds]
}
```

```
{Tue 8 Jul 2014 16:49:22, SandboxDate.png,
 <?xml version="1.0"?><api><error code="badupload_file" info="File upload param
   file is not a file upload; be sure to use multipart/form-data for your
   POST and include a filename in the Content-Disposition header." /></api>}
```

Attempt following http://mathematica.stackexchange.com/questions/52338/more-complete-mutipartdata-posts-using-urlfetch/97658#97658:

```
$WikiConnection@"edittoken"
```

```
1ebf41e07a1a3f585e57fc9b82d82c65+\
```

```
WikiUploadFile[name_String, description_String] :=
  WikiUploadFile[$WikiConnection, name, description];
WikiUploadFile[wc_WikiConnection, name_String, description_String] :=
  URLExecute[wc@"apiurl"; "http://requestb.in/1jorke91",
   "Method" → "POST",
   "Cookies" → wc@"cookies",
   "StoreCookies" → False,
   "MultipartElements" → {
     {"file\"; action=\"upload\"; text=\"" <> description <>
       "\"; format=\"xml\"; token=\"" <> (wc@"edittoken") <> "\\\"; filename=\"" <>
       name, "application/octet-stream", Import[name, "Byte"]}
    },
   "Headers" → {
     "Accept" → "application/json; charset=UTF-8",
     "Content-Type" → "multipart/form-data"
    }
  ];
{ds = DateString[],
 name = Export["SandboxDate.png", ds],
 WikiUploadFile[name, "SandboxDate.png by WikiLink2 on " <> ds]
}
```

```
{Sun 7 Feb 2016 16:42:37, SandboxDate.png, ok}
```