

```

WikiUserName::usage =
  "WikiUserName[] returns either the name of the user you are logged
  in as, your IP address if you're not logged in, or
  $Failed if something more complicated has happened!";

WikiAllPages::usage =
  "WikiAllPages[url,initialtext,namespace,namespaceNumber] attempts
  to produce a list of all pages in the namespace with
  titles beginning with initialtext. The first argument,
  url, shoould be the url of the wiki, ending in index.php.";

WikiPageMatchQ; WikiPageFreeQ; WikiStringReplace;
WikiStringCases; WikiPagesContaining;

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

CreateWikiConnection[baseURL_String] :=
  (InstallJava[]; GuessJarPath[];
   mediawikiConnection = JavaNew["wikilink.MediawikiConnection", baseURL];)

WikiConnectionValidQ[] :=
  If[JavaObjectQ[mediawikiConnection], True, If[defaultURL == "", False,
   CreateWikiConnection[defaultURL]; defaultURL = ""; WikiConnectionValidQ[]]]

WikiGetPageText::invalid =
  WikiSetPageText::invalid = "You must call CreateWikiConnection
  before using WikiGetPageText or WikiSetPageText";

(*WikifyName[name_String]:=StringReplace[name," ">"_"]*)

(*UnwikifyName[name_String]:=StringReplace[name,"_">" "]*)

WikiUploadFile[name_String, description_String] :=
  If[WikiConnectionValidQ[], mediawikiConnection@uploadFile[name, description],
   Message[WikiUploadFile::invalid]]

WikiUserName[] := Module[{text, cases},
  If[! WikiSetPageText["Who_Am_I", "----~"], Return[$Failed]];
  text = WikiGetPageText["Who_Am_I"];
  cases = StringCases[text, "--[User:" ~ x__ ~ "|" ~ x__ ~ "]" ~> x];
  If[Length[cases] == 1, Return[cases[[1]]];
  Return[$Failed];
]

WikiPageMatchQ[name_String, patt_] := StringMatchQ[WikiGetPageText[name], patt]

WikiPageMatchQ[names : {__String}, patt_] :=
  StringMatchQ[#[[2]], patt] & /@ WikiGetPageTexts[names]

WikiPagesContaining[names : {__String}, pattern_] :=
  Pick[names, Last /@ WikiGetPageTexts[names], _? (!StringFreeQ[#, pattern] &)]

WikiPageFreeQ[name_String, patt_] := StringFreeQ[WikiGetPageText[name], patt]

```

```

WikiPageFreeQ[names : {__String}, patt] :=
  StringFreeQ[#[[2]], patt] & /@WikiGetPageTexts[names]

WikiStringReplace[name_String, rules] :=
  WikiSetPageText[name, StringReplace[WikiGetPageText[name], rules]]

WikiStringReplace[name_String, rules, n_Integer] :=
  WikiSetPageText[name, StringReplace[WikiGetPageText[name], rules, n]]

WikiStringReplace[name_String, rules, description_String] :=
  WikiSetPageText[name, StringReplace[WikiGetPageText[name], rules], description]

WikiStringReplace[name_String, rules, n_Integer, description_String] :=
  WikiSetPageText[name, StringReplace[WikiGetPageText[name], rules, n], description]

WikiStringReplace[names : {__String}, rules] :=
  WikiSetPageTexts[{{#[[1]], StringReplace[#[[2]], rules]} & /@WikiGetPageTexts[names]}]

WikiStringReplace[names : {__String}, rules, n_Integer] := WikiSetPageTexts[
  {#[[1]], StringReplace[#[[2]], rules, n]} & /@WikiGetPageTexts[names]]

WikiStringReplace[names : {__String}, rules, summary_String] := WikiSetPageTexts[
  {#[[1]], StringReplace[#[[2]], rules]} & /@WikiGetPageTexts[names], summary]

WikiStringReplace[names : {__String}, rules, n_Integer, summary_String] :=
  WikiSetPageTexts[
  {#[[1]], StringReplace[#[[2]], rules, n]} & /@WikiGetPageTexts[names], summary]

WikiStringCases[name_String, rules] := StringCases[WikiGetPageText[name], rules]

WikiStringCases[names : {__String}, rules] :=
  {#[[1]], StringCases[#[[2]], rules]} & /@WikiGetPageTexts[names]

nextPageURL[hostname_, baseURL_, text_, initialText_: ""] :=
  Module[{candidates}, candidates = StringCases[text, "<a href=\"" ~
    nextURL : (baseURL ~ "?title=Special:Allpages&from=" ~ initialText ~
    ShortestMatch[___]) ~ "\" title=\"Special:Allpages\">Next page" =>
    "http://" <> hostname <> StringReplace[nextURL, "&" -> "&"], 1];
  If[Length[candidates] == 0, $Failed, candidates[[1]]]
]

```

```

WikiAllPages[hostname_String,
  baseURL_String: "/w/index.php", initialText_String: "",
  namespace_String: "", namespaceNumber_Integer: 0, maxPages_: ∞] :=
Module[{pages = {}, newPages, allPagesURL =
  "http://" <> hostname <> baseURL <> "?title=Special%3AAllpages&from=" <>
  initialText <> "&namespace=" <> ToString[nameSpaceNumber], allPagesText},
While[allPagesURL != $Failed ^ Length[pages] ≤ maxPages,
  Print["Looking at: " <> allPagesURL];
  allPagesText = Import[allPagesURL, "Text"];
  If[allPagesText == $Failed, Continue[]];
  allPagesURL = nextPageURL[hostname, baseURL, allPagesText, initialText];
  newPages =
    StringCases[allPagesText, "title=\"" ~ pagename : (((nameSpace ~ ":" | "") ~
      initialText ~ ShortestMatch[___]) ~ "\" :> pagename];
  Print["Found " <> ToString[Length[newPages]] <> " more pages."];
  pages = pages ~ Join ~ newPages;
];
pages
]

```