

MEDIAWIKI EXTENSION DEVELOPMENT

Richa Jain

OUTLINES

- Mediawiki
- What are Extensions?
- How to create Extensions?

What is Mediawiki ?

- Free and Open software Wiki package
- Written in PHP
- Wikipedia and other Wikis



Welcome to MediaWiki.org

MediaWiki is a [free software open source wiki](#) package written in [PHP](#), originally for use on [Wikipedia](#). It is now also used by several other projects of the non-profit [Wikimedia Foundation](#) and by [many other wikis](#), including this website, the home of MediaWiki.

Use the links below to explore the site contents. You'll find some content translated into other languages, but the primary documentation language is English.

For general questions about MediaWiki see the [communication page](#) or ask at the [support desk](#).

[About this site](#) | [About MediaWiki](#) | [Download](#) | [Help and support](#) | [Contribute](#)

- Main page
- Get MediaWiki
- Get extensions
- Communication
- Tech blog

- Support
 - User help
 - FAQ
 - Technical manual
 - Support desk

- Development
 - Bug tracker
 - Browse repository
 - Code docs
 - Bugs with patch
 - Statistics

- MediaWiki.org
 - Browse categories
 - Community portal
 - Recent changes
 - Current issues
 - Sandbox

- Print/export
 - Create a book
 - Download as PDF

Using MediaWiki



What is a wiki?

- Learn [how to navigate](#)
- Learn [how to edit a page](#)
- Get [more help](#)

System Administration



Install & configure MediaWiki

- [Upgrade](#) an existing MediaWiki installation
- Add features with [third-party extensions](#)
- Get [more sysadmin help](#)

Developing & Extending



Become a MediaWiki hacker

- Learn to use the [API](#) and [write extensions](#)
- Browse the [developer docs](#) and [class reference](#)
- Already a hacker? Visit the [developer hub](#)

Current version

- **1.23.0** – 2014-06-05

News

2014-06-05

New opportunities

2014-06-10

WHAT SHOULD YOU HAVE?

- PHP, JavaScript/CSS
- MySQL (orSQLite)
- SublimeText, Geany, Vim or Notepad++

Mediawiki Installation

- Requirements
- Download it
- Put it in your localhost
- Open it in a web browser and follow the instructions to install



MediaWiki 1.22alpha

LocalSettings.php not found.

Please [set up the wiki](#) first.

Set \$wgLogo to the URL path to your own logo image.

Page Discussion

Read Edit View history View annotations Search Go Search

Main Page

MediaWiki has been successfully installed.

Consult the [User's Guide](#) for information on using the wiki software. hihi

Getting started [\[edit\]](#)

- [Configuration settings list](#)
- [MediaWiki FAQ](#)
- [MediaWiki release mailing list](#)
- [Localise MediaWiki for your language](#)

Intro

This page was last modified on 25 June 2014, at 06:36.

This page has been accessed 2,977 times.

[Privacy policy](#) [About media](#) [Disclaimers](#)

Navigation
Main page
Recent changes
Random page

Toolbox
What links here
Related changes
Special pages
Permanent link
Page information

Print/export
Create a book
Download as PDF
Download as DocBook
XML
Printable version



What are Extensions?

- Extend the Wiki markup
- Add new reporting and administrative features
- Change the look and design

HelloUser/HelloUser.php

HelloUser/HelloUser.i18n.php

HelloUser/HelloUser.body.php

Developing Extensions

- Setup (stores the setup instructions)
- Execution (execution code for the extension)
- Localisation

Setup

```
HelloUser.php x HelloUser.i18n.php x HelloUser.body.php x HelloUser.php x HelloUser.hooks.php x
<?php

$wgExtensionCredits['other'][] = array(
    'path' => __FILE__,
    'name' => 'HelloUser',
    'version' => '0.1',
    'author' => array( 'Richa Jain' ),
    'url' => '',
);

$dir = dirname( __FILE__ ) . '/';

$wgExtensionMessagesFiles['HelloUser'] = $dir . 'HelloUser.i18n.php';

Saving file '/home/richa/Desktop/HelloUser/HelloUser.php'...
PHP Tab Width: 8 Ln 15, Col 1 INS
```

Installing Extensions

- Include the Setup file in Localsettings.php
- You are done installing the extension!
- Head on towards Internationalization

Internationalisation

HelloUser.php x HelloUser.i18n.php x

```
<?php
/* message file for the extension
 *
 */
$messages = array();

$messages['en'] = array(
    'example-hello' => 'Hello User',
    'example-helloworld-intro' => 'Intro'
);
```

Using Messages

```
wfMessage('message-name')->parse()
```

```
wfMessage('message-name')->plain()
```

```
wfMessage('message-name', param1, param2)->parse()
```

Extension Types

- Special Pages
- Hooks
- Tag Extensions

Special Page Setup

```
HelloUser.php x HelloUser.i18n.php x HelloUser.body.php x HelloUser.php x HelloUser.hooks.php x
<?php

$wgExtensionCredits['other'][] = array(
    'path' => __FILE__,
    'name' => 'HelloUser',
    'version' => '0.1',
    'author' => array( 'Richa Jain' ),
    'url' => '',
);

$dir = dirname( __FILE__ ) . '/';

$wgExtensionMessagesFiles['HelloUser'] = $dir . 'HelloUser.i18n.php';

$wgSpecialPages['HelloUser'] = 'SpecialHelloUser'; //Tells mediawiki about the new special page and its classname
$wgAutoloadClasses['SpecialHelloUser'] = $dir . 'HelloUser.body.php'; //Location of the new special class
```

Special Page File

```
HelloUser.php x HelloUser.i18n.php x HelloUser.body.php x
<?php
class SpecialHelloUser extends SpecialPage {
    function __construct() {
        parent::__construct( 'HelloUser' ); //This names your special page
    }

    function execute( $par ) { //If some migrates to special:extension/blah, $par will contain blah

        $out = $this->getOutput(); //Returns the output page object
        $title = wfMessage('example-hello'); //extract message from the i18n file
        $out->setPageTitle( $title ); //set the special page's title
    }
}
```

Hooks Setup

```
HelloUser.php x HelloUser.i18n.php x HelloUser.body.php x HelloUser.php x HelloUser.hooks.php x
<?php

$wgExtensionCredits['other'][] = array(
    'path' => __FILE__,
    'name' => 'HelloUser',
    'version' => '0.1',
    'author' => array( 'Richa Jain' ),
    'url' => '',
);

$dir = dirname( __FILE__ ) . '/';

$wgExtensionMessagesFiles['HelloUser'] = $dir . 'HelloUser.i18n.php';

$wgSpecialPages['HelloUser'] = 'SpecialHelloUser'; //Tells mediawiki about the new special page and its classname
$wgAutoloadClasses['SpecialHelloUser'] = $dir . 'HelloUser.body.php'; //Location of the new special class

$wgHooks['BeforePageDisplay'][] = 'HelloUserHooks::onBeforePageDisplay'; //Registers the Hook event
$wgAutoloadClasses['HelloUserHooks'] = $dir . 'HelloUser.hooks.php'; //Location of the class name
```

```
<?php
/*Extension Hooks */

class HelloUserHooks {
    public static function onBeforePageDisplay(OutputPage &$output) {

        /*define your function*/
        $output->addWikiText('Hello');
    }
}
```

Parser Setup

HelloUser.php x HelloUser.i18n.php x HelloUser.body.php x HelloUser.php x HelloUser.hooks.php x

```
<?php

$wgExtensionCredits['other'][] = array(
    'path' => __FILE__,
    'name' => 'HelloUser',
    'version' => '0.1',
    'author' => array( 'Richa Jain' ),
    'url' => '',
);

$dir = dirname( __FILE__ ) . '/';

$wgExtensionMessagesFiles['HelloUser'] = $dir . 'HelloUser.i18n.php';

$wgSpecialPages['HelloUser'] = 'SpecialHelloUser'; //Tells mediawiki about the new special page and its classname
$wgAutoloadClasses['SpecialHelloUser'] = $dir . 'HelloUser.body.php'; //Location of the new special class

$wgHooks['BeforePageDisplay'][] = 'HelloUserHooks::onBeforePageDisplay'; //Registers the Hook event
$wgAutoloadClasses['HelloUserHooks'] = $dir . 'HelloUser.hooks.php'; //Location of the class name

$wgHooks['ParserFirstCallInit'][] = 'HelloUserHooks::sampleParserInit';
```

```
<?php
/*Extension Hooks */

class HelloUserHooks {
    public static function onBeforePageDisplay(OutputPage &$output) {

        /*define your function*/
        $output->addWikiText('Hello');

    }

    public function sampleParserInit(Parser $parser) {
        //when the parser just sees the <hello> tag, it executes the sampleRender function described below
        $parser->setHook('hello', sampleRender);
        return true;
    }

    public function sampleRender($input, array $arg, Parser $parser, PPFrame $frame ) {
        //$input is the input between <hello> and </hello> tags
        //$arg is an array of the tag arguments passed with <hello arg1="" arg2="">
        //$parser is the parser object
        return htmlspecialchars($input);
    }
}
```

Thank You