

---+ Package TWiki::Plugins

This module defines the singleton object that handles Plugins loading, initialization and execution.

This class uses Chain of Responsibility (GOF) pattern to dispatch handler calls to registered plugins.

Note that as of version 1.026 of this module, TWiki internal methods are *no longer available* to plugins. Any calls to TWiki internal methods must be replaced by calls via the \$SESSION object in this package, or via the Func package. For example, the call:

```
my $pref = TWiki::getPreferencesValue('URGH');
```

should be replaced with

```
my $pref = TWiki::Func::getPreferencesValue('URGH');
```

and the call

```
my $t = TWiki::writeWarning($message);
```

should be replaced with

```
my $pref = $TWiki::Plugins::SESSION->writeWarning($message);
```

Methods in other modules such as Store must be accessed through the relevant TWiki sub-object, for example

```
TWiki::Store::saveTopic(...)
```

should be replaced with

```
$TWiki::Plugins::SESSION->{store}->saveTopic(...)
```

Note that calling TWiki internal methods is very very bad practice, and should be avoided wherever practical.

The developers of TWiki reserve the right to change internal methods without warning, unless those methods are clearly marked as PUBLIC. PUBLIC methods are part of the core specification of a module and can be trusted.

PUBLIC constant \$VERSION

This is the version number of the plugins package. Use it for checking if you have a recent enough version.

PUBLIC \$SESSION

This is a reference to the TWiki session object. It can be used in plugins to get at the methods of the TWiki kernel.

You are *highly* recommended to only use the methods in the Func interface, unless you have no other choice, as kernel methods may change between TWiki releases.

ClassMethod new (\$session)

Construct new singleton plugins collection object. The object is a container for a list of plugins and the handlers registered by the plugins. The plugins and the handlers are carefully ordered.

ObjectMethod finish ()

Break circular references.

ObjectMethod load (\$allDisabled) -> \$loginName

Find all active plugins, and invoke the early initialisation. Has to be done *after* prefs are read.

Returns the user returned by the last `initializeUserHandler` to be called.

If `allDisabled` is set, no plugin handlers will be called.

ObjectMethod settings ()

Push plugin settings onto preference stack

ObjectMethod enable ()

Initialisation that is done after the user is known.

ObjectMethod getPluginVersion () -> \$number

Returns the `$TWiki::Plugins::VERSION` number if no parameter is specified, else returns the version number of a named Plugin. If the Plugin cannot be found or is not active, 0 is returned.

ObjectMethod addListener (\$command, \$handler)

- `$command` - name of the event
- `$handler` - the handler object.

Add a listener to the end of the list of registered listeners for this event. The listener must implement `invoke($command, ...)`, which will be triggered when the event is to be processed.

ObjectMethod dispatch (\$handlerName, ...)

Dispatch the given handler, passing on ... in the parameter vector

ObjectMethod haveHandlerFor (\$handlerName) -> \$boolean

- `$handlerName` - name of the handler e.g. `preRenderingHandler`

Return: true if at least one plugin has registered a handler of this type.

This topic: TWiki > TWikiPluginsDotPm

Topic revision: r4 - 2010-05-29 - TWikiContributor



Copyright © 1999-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback

Note: Please contribute updates to this topic on TWiki.org at TWiki:TWiki.TWikiPluginsDotPm.