Конфигурация

DokuWiki позволяет создавать очень гибкие плагины и шаблоны делая их легко настраиваемыми..

Default Settings

Чтобы сделать плагин или шаблон настраиваемым, необходимо предоставить `lib/plugins/<plugin>/conf/default.php` который будет содержать настройки по умолчанию, и `lib/plugins/<plugin>/conf/metadata.php` который содержит описывающие метаданные конфигурации используемые диспетчером конфигурации для обработки/отображения параметров [1].

```
$conf[<setting>] = <value>;
```

Конфигурационные метаданные

Для каждого параметра `lib/plugins/<your plugin>/conf/default.php` должно быть `$meta[<setting>]` значение, определенное в `lib/plugins/<your plugin>/conf/metadata.php`[2]:

```
$meta[<setting>] = array(<setting class>, <param name> => <param value>);
```

Если для класса настроек не требуется никаких параметров (см. ниже), то это просто:

```
$meta[<setting>] = array(<setting class>);
```

Примеры:

```
$meta['_basic']     = array('fieldset');
$meta['title']      = array('string');
$meta['lang']       = array('dirchoice','_dir' => DOKU_INC.'inc/lang/');
$meta['dmode']      = array('numeric','_pattern' => '/0[0-7]{3,4}/'); //
only accept octal representation
$meta['allowdebug'] = array('onoff');
$meta['passcrypt']  = array('multichoice','_choices' =>
array('smd5','md5','sha1','ssha','crypt','mysql','my411'));
```

Классы

`''`	DKласс по умолчанию («настройка»), текстовая область, минимальная проверка входных данных, вывод настроек в кавычках.
`'string'`	Ввод текста в одну строку, минимальная проверка ввода, вывод в кавычках.

'numeric'	Ввод текста, принимает числа и арифметические операторы, установка вывода без кавычек. Если заданы параметры `_min` и `_max` используются для проверки.
'numericopt'	То же, что и выше, но принимает пустые значения.
'onoff'	Входные данные флажка, выходные данные настройки `0` или `1` (обратите внимание, что значения по умолчанию должны быть одним из этих целых чисел, а не логическим значением)..
'multichoice'	Выберите входные данные (единственный выбор), установите выходные данные в кавычках, обязательный `_choices` параметр.
'email'	Ввод текста, ввод должен соответствовать формату адреса электронной почты, вывод должен быть заключен в кавычки.
'password'	Ввод пароля, минимальная проверка ввода, настройка вывода обычного текста в кавычках. После установки пароль больше не отображается в менеджере конфигурации. Может быть запутан в конфигурации через `_code` параметр.
'dirchoice'	Как `multichoice`, варианты выбора основаны на папках, найденных в месте, указанном в `_dir` параметре (обязательно).
'multicheckbox'	Флажок для каждого варианта плюс строковый ввод «другое», настройка файла конфигурации представляет собой список отмеченных вариантов, разделенных запятыми.
'fieldset'	Используется для группировки настроек конфигурации, но сам по себе не является настройкой. Чтобы это было понятно в языковых файлах, ключи для этого типа должны начинаться с `_`.
'authtype'	Создает выборку доступных методов аутентификации на основе имен классов, `inc/auth` соответствующих шаблону `authtype.class.php`.
'array'	простой (одномерный) массив строковых значений, отображаемый в виде списка, разделенного запятыми, в диспетчере конфигурации, но сохраняемый как PHP array(). Значения не могут содержать запятые. `_patter` nподдерживается сопоставление значений массива.
'regex'	Строка регулярного выражения, обычно без разделителей; как и для `string`, дополнительно протестировано, чтобы увидеть, будет ли компилироваться и запускаться как регулярное выражение. В дополнение к `_pattern`, также принимает `_delimiter` (default `/`) и `_pregflags` (default `ui`).

Параметры

'_pattern'	Строка, регулярное выражение. Ввод проверяется на соответствие этому шаблону перед принятием. Необязательно для всех классов, кроме `onoff`, `multichoice` и `dirchoice` которые его игнорируют.
'_choices'	Массив вариантов выбора. Используется для заполнения поля выбора. Выбор будет заменен локализованной языковой строкой, индексированной по `<setting name>_o_<choice>`, если она существует. Требуется для `multichoice` & `multicheckbox` классов, игнорируется другими.
'_dir'	Расположение каталога, который будет использоваться для заполнения списка выбора. Требуется `dirchoice` классом, игнорируется другими классами.
'_code'	Устанавливает опцию запутывания для полей пароля. Может быть `plain`, `base64` или `uuencode`. При использовании последних двух вам необходимо использовать conf_decodeString() для доступа к простому значению..

'_combine'	Дополнительные выходные значения настроек, которые можно объединить в один дисплей checkbox. Необязательно для multicheckbox, игнорируется другими классами.
'_code'	Метод кодирования, допустимые значения: base64, uuencode, plain. По умолчанию — обычный.
'_min'	Минимальное числовое значение. Необязательно для numeric и numericopt, игнорируется другими.
'_max'	Максимальное числовое значение. Необязательно для numeric и numericopt, игнорируется другими.
'_delimiter'	Строка, по умолчанию / — один символ, используемый в качестве разделителя для проверки входных значений регулярного выражения.
'_pregflags'	Строка, по умолчанию ui, допустимые модификаторы шаблона preg, используемые при тестировании входных значений регулярных выражений, для получения дополнительной информации см. http://uk1.php.net/manual/en/reference.pcre.pattern.modifiers.php.
'_multiple'	Bool, разрешить несколько значений email, разделенных запятыми. Необязательно для email, игнорируется другими.
'_other'	Как обрабатывать другие значения (не перечисленные в _choices). Принятые значения: always, exists, never. Значение по умолчанию always. exists показывает поле ввода «другое», только если настройка содержит значение(я), не перечисленное в вариантах (например, из-за ручного редактирования или обновления, изменяющегог _choices). Это безопаснее, чем never так как не отбрасывает неизвестные/другие значения. Необязательно для multicheckbox, игнорируется другими.

Примеры

"multicheckbox" с "_other"

Предположим, изначально плагин использует следующий синтаксис:x:

```
$meta['multi'] = array('multicheckbox', '_choices' =>
array('a','b','c','d'));
```

это означает, что в поле с несколькими флажками будут отображаться четыре варианта выбора и строковое поле ввода.

Предположим также, что пользователь вставляет одно или несколько дополнительных значений, отсутствующих в _choices.

В следующем выпуске плагина авторский плагин решает использовать multicheckbox без дополнительной строки, но, в целях обратной совместимости, принимает:

```
$meta['multi'] = array('multicheckbox', '_other' => 'exists', '_choices' =>
array('a','b','c','d'));
```

что означает:

- если пользователь ввел значения, разделенные запятыми, они будут напечатаны в дополнительной строке;
- в противном случае дополнительная строка не будет отображаться.

Обратите внимание также, что если пользователь вставляет разделенные запятыми значения, которые уже существуют (или некоторые из них), в `_choices`:

- они уже отмечены, то лишняя строка будет удалена и ничего больше;
- если они еще не отмечены, то лишняя строка будет удалена и отмечено относительное значение.

Доступ к настройкам

Основные настройки

Внутри init.php настройки конфигурации считываются в глобальный массив `$conf[]`. Если настройки не заданы, они считываются из файла настроек по умолчанию. Вы можете получить доступ к основным настройкам в любом месте, используя `$conf[]` массив.

```
$startpage = $conf['start'];
```

Настройки плагина

Вы можете получить доступ к настройкам плагина используя `$this->getConf('<setting>')` метод. В классе вашего плагина используете:

```
$keyvalue = $this->getConf('key');
```

Настройки шаблона

В шаблонах можно использовать `tpl_getConf('<setting>')`.

```
$nicetoknow = tpl_getConf('special');
```

Метки в диспетчере конфигураций

Для каждого параметра в `lib/plugins/<your plugin>/conf/default.php` может быть `$lang[<setting>]` определено значение в `lib/plugins/<your plugin>/lang/en/settings.php`. Это значение будет отображаться как метка параметра в диспетчере конфигурации. Если файл метки пропущен или не содержит значения для параметра, диспетчер конфигурации отобразит «`plugin <plugin name> <setting>`» в качестве метки..

Вы также можете создать `settings.php` файл для других языков.

Опять же, это относится и к шаблонам (более подробную информацию см. в Локализация).

1)
шаблоны аналогичны lib/tpl/<template>/conf/default.php и т.д.

[2)]

Для шаблонов все полностью аналогично: lib/tpl/<your template>/conf/default.phpи lib/tpl/<your template>/conf/metadata.php.

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=wiki:devel:configuration

Last update: **2025/01/03 16:55**