Система управления контентом SantaFox™

Руководство по API SantaFox

Версия 1.0

SantaFox developer
Saint-Peterburg, 2008

Оглавление

Введение ... 3
Для кого это руководство ... 3
Правила группирования функций ... 3
Функции ядра (проверенные) ... 4
Прямой доступ к методам ... 4
Сохранение произвольных значений ... 6
Регистрация и авторизация пользователей ... 8
Работа с файлами и папками ... 18
Сбор данных о посетителе ... 24
Управление административным интерфейсом ... 27
Функции взаимодействия ... 39
Функции обработки данных ... 53
Заключение ... 57

Введение

Для кого это руководство

Данное руководство является продолжением руководства по разработке модулей и даёт перечень функций ядра и ряда других элементов SanataFox, которые Вы можете использовать при разработке собственных модулей.

Сначала все доступные функции разбиты по группам, в зависимости от цели их использования. Кроме этого вы можете воспользоваться алфавитным указателем этих функций для поиска описаний к конкретной функции.

Правила группирования функций

Функции ядра

Прямой доступ к методам

Собранные функции используются только в тех методах модуля, к которым открыт прямой доступ (Direct Access, DA). Прямой доступ к методам позволяет через GET (или POST) запрос вызвать на выполнение какой-либо публичный метод модуля.

При этом, такой вызов может быть либо сквозным, либо конечным. При сквозном вызове, после передачи управления указанному методу, управление будет возвращено ядру и будет произведено формирование страницы. Конечный вызов, закончит формирование страницы сразу после того как будет вызван публичный метод ядра, а возвращенный им результат будет выведен на экран.

Сейчас DA имеет не богатые функциональные возможности, но постепенно набор доступных функций будет увеличиваться.

Перечень всех методов

pub_da_link_createpub_session_get	Создаёт URL для вызова метода через DA.
pub_da_metod_set	Задаёт имя метода модуля, участвующего в прямом вызове.
pub_da_page_template_set	Подменяет шаблон страницы сайта.

pub_da_link_create

$kernel → pub_da_link_create ($metod_name)

Назначение

Создаёт URL для вызова метода через DA.

Параметры

- $metod_name: *String*
  Имя метода модуля, который должен быть вызван.

Возвращаемое значение

- *String*
  URL, запрос по которому приведёт к вызову метода.

Описание

Метод используется в публичных методах модуля (как правило), для формирования URL, по которому сможет перейти пользователь.

Другие функции

pub_da_metod_set

$kernel → pub_da_metod_set ($id_metod, $run_and_stop = true)

Назначение

Задаёт имя метода модуля, участвующего в прямом вызове.

Параметры

- $id_metod: *String*
  Имя метода, как он назван в классе, который можно вызывать прямым методом.
- $run_and_stop: *Bool*
  Если *TRUE*, то после отработки метода его результат будет выведен на экран и дальнейшей обработки не будет (конечный метод). В противном случае, после работы метода, страница будет строиться дальше, по своему алгоритму (сквозной метод).

Возвращаемое значение

- *Bool*
  Возвращается *true*, если метод успешно добавлен к списку методов, доступных к прямому вызову.

Описание

Используется, как правило, в конструкторе класса. Этим методом вы сообщаете ядру о тех модулях, к которым можно будет обратиться средствами DA, а кроме того, указываете, прерывать ли обработку страницы после выполнения такого метода или нет.

Пример:

```
class pageprint
{

   function pageprint()
   {
      global $kernel;

      $kernel->pub_module_open_metod_set('pub_metod', false);
   }

   function pub_metod()
   {
      global $kernel;

      $html = "";
      //...
      return $html;
   }
```

```
}
```

Другие функции

pub_da_page_template_set

$kernel → pub_da_page_template_set ($file_name)

Назначение

Подменяет шаблон страницы сайта.

Параметры

- $file_name: *String*
  Путь и файл с новым шаблоном страницы.

Возвращаемое значение

- *Void*

Описание

Функция применяется только в методах модулей, вызываемых средствами DA (Direct Access).

После вызова функции, шаблон страницы заменяется на указанный. И когда метод (модуля) закончит свою работу и вернёт управление классу, строящему страницу, в качестве шаблона страницы, будет использоваться указанный файл. Соответственно будут обработаны и выведены только те метки, которые указаны в этом шаблоне.

Другие функции

Сохранение произвольных значений

Собраны функции, которые позволяют модулю сохранять какие-то временные значения. Как правило, данные методы удобно использовать для временного хранения данных, полученных через POST и GET запросы, с тем что бы передавать их между разными страницами сайта.

Перечень всех методов

pub_session_get	Возвращает сохраненное значение из сессии.
pub_session_set	Сохраняет значение в сессию.
pub_session_unset	Удаляет все (или конкретные) сохранённые в сессии значения.

pub_session_get

$kernel → pub_session_get ($name = null)

Назначение

Возвращает сохраненное значение из сессии.

Параметры

- $name: *String*
  Имя получаемого значения.

Возвращаемое значение

- *String* | *Int* | *Array*

Описание

Функция возвращает сохраненные в сессии значения. Для получения значения необходимо указать имя, под которым значение было сохранено. Если имя не указано, то будет возращён весь массив сохраненных переменных.
Следует учитывать, что область видимости сохраненных значений у каждого модуля своя.

Другие функции

pub_session_set, pub_session_unset

pub_session_set

$kernel → pub_session_set($name, $value = '')

Назначение

Сохраняет значение в сессию.

Параметры

- $name: *String*
  Имя сохраняемого значения.
- $value: *String* | *Int* | *Array*
  Само сохраняемое значение.

Возвращаемое значение

- *Bool*

Описание

Функция сохраняет в сессии какое-то значение, доступ к которому в дальнейшем можно получить с помощью метода pub_session_get. Для сохранения значения необходимо указать его имя и непосредственное само сохраняемое значение. В случае успешной установки возвращает true, в случае неудачи – false.
Следует учитывать, что область видимости сохраненных значений у каждого модуля своя.

Другие функции

pub_session_get, pub_session_unset

pub_session_unset

$kernel → pub_session_unset ($name = null, $module_id = null)

Назначение

Удаляет все (или конкретные) сохранённые в сессии значения.

Параметры

- $name: *String*
  Имя удаляемого значения.
- $module_id: *String*
  Идентификатор модуля, чьи значений будут удаляться.

Возвращаемое значение

- *Bool*

Описание

Функция очищает все значения, которые были сохранены модулем. Если задан параметр $name то удаляется только одно конкретное значение. Второй параметр можно использовать для очистки сохраненных значений другого модуля.

Другие функции

pub_session_get, pub_session_unset

Регистрация и авторизация пользователей

В этом разделе собраны методы ядра, которые отвечают за авторизацию и регистрацию посетителей сайта. Большинством указанных здесь функций пользуется модуль авторизации пользователей, часть функций используются и другими модулями, что бы получить какие-либо данные об авторизированном пользователе.

Следует отметить, что весь вопрос управления пользователями осуществляется ядром CMS. Модулю авторизации отводится роль по формированию административного интерфейса, для реализации этого управления, а также роль формирования форм авторизации и регистрации непосредственно для посетителя сайта.

Перечень всех методов

pub_user_add_new	Метод для вывода отладочной информации.
pub_user_change_enabled	Включает или отключает пользователя фронт-офиса.
pub_user_delete	Производит полное удаление пользователя сайта из базы.
pub_user_field_get	Возвращает значение конкретного поля у текущего

	(авторизированного) пользователя.
pub_user_group_get	Получает информацию о группах пользователя сайта.
pub_user_info_get	Возвращает всю доступную информацию о текущем (авторизированном) пользователе фронт-офиса.
pub_user_is_registred	Проверяет, авторизирован текущий пользователь в системе или нет.
pub_user_login_info_get	Возвращает всю доступную информацию о пользователе по логину.
pub_user_register	Производит авторизацию пользователя сайта.
pub_user_unregister	Очищает информацию о текущем пользователе сайта.
pub_user_verify	Подтверждает учетную запись пользователя сайта по переданному идентификатору пользователя.
pub_users_fields_get	Возвращает массив с дополнительными полями, которые добавили модули.
pub_users_group_get	Возвращает список всех доступных групп пользователей сайта.
Ошибка! Источник ссылки не найден.	Сохраняет группы, в которые входит пользователь.
pub_users_info_get	Возвращает всю доступную информацию о пользователе (пользователях) сайта.
pub_users_info_set	Записывает измененную информацию о пользователе.

pub_user_add_new

$kernel → pub_user_add_new ($login, $password, $email, $name, $unic_login = true)

Назначение

Метод для вывода отладочной информации.

Параметры

- $login: *String*
- $password: *String*
- $email: *String*
- $name: *String*
- $unic_login: *Bool*
  Если *true* – то уникальность пользователя проверяется по логину, если *false* – то по email-y.

Возвращаемое значение

- *Int*

Описание

Функция добавляет нового пользователя сайта. В качестве параметров передаются данные о новом пользователе. В качестве уникального идентификации пользователя может быть использован либо логин, либо адрес электронной почты.

Возвращает идентификатор вновь добавленного пользователя либо код ошибки (с отрицательным знаком). Возможны следующие варианты ошибки:
-1 : пользователь с таким логином (email-ом) уже существует и НЕ подтвержден.
-2 : пользователь с таким логином (email-ом) уже существует и подтвержден.

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации.

Другие функции

pub_user_change_enabled

$kernel → pub_user_change_enabled ($id_user, $enabled = true)

Назначение

Включает или отключает пользователя фронт-офиса.

Параметры

- $id_user: *Int*
  Идентификатор пользователя сайта.
- $ enabled: *Bool*
  Определяет включить или отключить учётную запись.

Возвращаемое значение

- *Bool*

Описание

Функция используется для того, чтобы можно было, например, временно отключить кого-то из посетителей

Возвращает *true* - если действие выполнено успешно.

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_user_delete

$kernel → pub_user_delete ()

Назначение

Производит полное удаление пользователя сайта из базы.

Параметры

- $id_user: *Int*
  Идентификатор пользователя сайта, которого нужно удалить.

Возвращаемое значение

- *Bool*

Описание

Производит полное удаление всей информации пользователе сайта.

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_user_field_get

$kernel → pub_user_field_get ($name_field, $name_modul = "")

Назначение

Возвращает значение конкретного поля у текущего (авторизированного) пользователя.

Параметры

- $name_field: *String*
  Имя идентификатора свойства.
- $name_modul: *String*:
  Идентификатор модуля, если это необходимо.

Возвращаемое значение

- *Array*

Описание

Функция используются (как правило) теми модулями, которые добавляли какие-то дополнительные поля к пользователям сайта, и хотят получить эти значения.

Второй параметр может использоваться в том случае, если необходимо узнать значение свойства к пользователю сайта, добавленное другим (не текущим модулем).

Другие функции

pub_user_group_get

$kernel → pub_user_group_get ($id)

Назначение

Получает информацию о группах пользователя сайта.

Параметры

- $id: *Int*
  Идентификатор пользователя, чьи данные запрашиваются.

Возвращаемое значение

- *Array*

Описание

Функция возвращает массив групп, к которым принадлежит пользователь сайта.

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_user_info_get

$kernel → pub_user_info_get ($tree = false)

Назначение

Возвращает всю доступную информацию о текущем (авторизированном) пользователе фронт-офиса.

Параметры

- $tree: *Bool*
  Определяет как формировать параметры, добавленные модулями.

Возвращаемое значение

- *Array*

Описание

Возвращаемый массив может иметь два вида – линейный и древовидный (в зависимости от передаваемого в метод параметра). Если передано *true*, то параметры будут сгруппированы по модулям, их добавившим, в противном случае, всё будет представлено в линейном виде.

Другие функции

pub_users_info_get

pub_user_is_registred

$kernel → pub_user_is_registred ()

Назначение

Проверяет, авторизирован текущий пользователь в системе или нет.

Параметры

Отсутствуют.

Возвращаемое значение

- *Bool*

Описание

Функция проверяет, авторизирован ли текущий пользователь сайта или нет.

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_user_login_info_get

$kernel → pub_user_login_info_get ($login, $is_login = true)

Назначение

Возвращает всю доступную информацию о пользователе по логину.

Параметры

- $login: *String*
  Логин или e-mail пользователя, чьи данные запрашиваются.
- $is_login: *Bool*
  Если *true* - то первый параметр логин, в противном случае первый параметр - e-mail.

Возвращаемое значение

- *Array*

Описание

В качестве параметра передается логин пользователя или e-mail, по которому будет определён пользователь.

Другие функции

pub_user_info_get**,** pub_users_info_get

pub_user_register

$kernel → pub_user_register ($login, $password, $unic_login = true)

Назначение

Производит авторизацию пользователя сайта.

Параметры

- $login: *String*
  Передаётся контент для вывода.
- $password: *String*
- $unic_login: *Bool*
  Если *true* - то содержимое параметра login воспринимается как логин пользователя, в противном случае, подразумевается что там e-mail.

Возвращаемое значение

- *Int*

Описание

Функция производит авторизацию пользователя сайта по переданному логину (или e-mail`y).

Регистрация происходит по переданному логину и паролю и делает этого пользователя текущим. В качестве логина может быть передан как непосредственно логин так и e-mail.

Варианты возвращаемого значения:

1 : пользователь с таким логином (email-ом) успешно авторизирован.
-1 : пользователь с таким логином (email-ом) не существует.
-2 : пользователь с таким логином (email-ом) отключен администратором сайта
-3 : пользователь с таким логином (email-ом) не подтвердил свою регистрацию.

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации.

Другие функции

pub_user_unregister

$kernel → pub_user_unregister()

Назначение

Очищает информацию о текущем пользователе сайта.

Параметры

Отсутствуют.

Возвращаемое значение

- *Void*

Описание

Осуществляет де-авторизацию пользователя сайта.

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_user_verify

$kernel → pub_user_verify ($id_user)

Назначение

Подтверждает учетную запись пользователя сайта по переданному идентификатору пользователя.

Параметры

- $id_user: *Int*
  Идентификатор пользователя сайта.

Возвращаемое значение

- *Bool*
  Возвращает *true* если подтверждение прошло успешно и *false* в противном случае.

Описание

Изначально зарегистрированный пользователь находится в неподтвержденном состоянии, то есть учетная запись уже существует, но для того, чтобы ей можно было пользоваться, необходимо её подтвердить (обычно по e-mail`у).

Другие функции

pub_users_fields_get

$kernel → pub_users_fields_get ()

Назначение

Возвращает массив с дополнительными полями, которые добавили модули.

Параметры

Отсутствуют.

Возвращаемое значение

- *Array*

Описание

Дополнительные поля, это те поля, которые были прописаны модулями. Возвращается просто массив доступных полей, без значений у конкретных пользователей.

Другие функции

pub_users_group_get

$kernel → pub_users_group_get ()

Назначение

Возвращает список всех доступных групп пользователей сайта.

Параметры

Отсутствуют.

Возвращаемое значение

- *Array*

Описание

Возвращает массив вида:

```
[0] => Array
 (
    [id] => 1
    [name] => standart
    [full_name] => Обычные посетители
 )
```

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_users_group_set

$kernel → pub_users_group_set ($id, $data)

Назначение

Сохраняет группы, в которые входит пользователь.

Параметры

- $id: *Int*
  Идентификатор пользователя, чьи данные сохраняются.
- $data: *Array*
  Массив групп, к которым относится пользователь.

Возвращаемое значение

- *Bool*

Описание

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_users_info_get

$kernel → pub_users_info_get ($id_user = "", $tree = true)

Назначение

Возвращает всю доступную информацию о пользователе (пользователях) сайта.

Параметры

- $id_user: *Int*
  Идентификатор пользователя сайта (используемый в mySql).
- $tree: *Bool*
  Определяет то, как группировать параметры, добавленные модулями к пользователю (пользователям) сайта.

Возвращаемое значение

- *Array*

Описание

Возвращает массив со всеми пользователями (или каким-то конкретным) и их данными.
Возвращаемый результат идентичен тому, который возвращает функция pub_user_info_get

Другие функции

pub_user_info_get**,** pub_user_login_info_get

pub_users_info_set

$kernel → pub_users_info_set ($data, $update_curent = true)

Назначение

Записывает измененную информацию о пользователе.

Параметры

- $data: *Array*
  Массив с данными пользователя, подлежащие сохранению. Массив должен быть идентичен тому, который возвращает функция pub_users_info_get.
- $update_curent: *Bool*
- Если *true*, то данные будут не просто записаны, но и сразу обновлены в сессии.

Возвращаемое значение

- *Array*

Описание

Данная функция используются модулем авторизации, но может быть использована и другими модулями, подменяющими собой функции модуля авторизации, или для других целей.

Другие функции

pub_users_info_get

Работа с файлами и папками

В этом разделе собраны все функции, которые облегчат взаимодействие модуля с файловой системой сервера.

Перечень всех методов

pub_dir_create_in_files	Создание новой директории в папке /content.
pub_dir_recurs_delete	Удаляет заданную директорию и всё её содержимое.
pub_files_list_get	Получить массив файлов в указанном каталоге.
pub_image_save	Меняет размеры загруженной картинки до нужных размеров и сохраняет ее в каталог.
pub_image_text_write	Добавляет текст поверх изображения.
pub_site_root_get	Возвращает путь к корню сайта.

pub_dir_create_in_files

$kernel → pub_dir_create_in_files ($dir = false, $direct = false)

Назначение

Создание новой директории в папке /content.

Параметры

- $dir: *String*
  Строка пути, который необходимо создать.
- $direct: *Bool*
  Если установлен в true то указанный в путь ($dir) рассматривается от корня сайта. По умолчанию, указанный путь создаётся в папке content/files.

Возвращаемое значение

- *Bool*

Описание

Чаще всего метод используются модулями, для того что бы создать необходимые для себя папки, в которых могут храниться какие-то файлы. Как правило, метод вызывается в инсталляторах модулей. В качестве параметра разделителя директорий может быть передан как символ кассой черты, так и символ вертикальной черты и обратной кассой черты («/», «|», «\»)

Пример:

```
function install_children($id_module)
{
  global $kernel;

  $kernel->pub_dir_create_in_files($id_module.'|filedoc');
  $kernel->pub_dir_create_in_files($id_module.'|filepdf');
 }
```

В результате такого вызова будут созданы следующие папки (допустим, что $id_module = 'news'):

```
/content/files/news
/content/files/news/filedoc
/content/files/news/filepdf
```

Другие функции

pub_dir_recurs_delete

$kernel → pub_dir_recurs_delete ($dir, $no_delet_main = false)

Назначение

Удаляет заданную директорию и всё её содержимое.

Параметры

- $dir: *String*

Путь к папке, чьё содержимое необходимо узнать.
- $no_delet_main: *Bool*

Возвращаемое значение

- *Void*

Описание

Удаляются как файлы, так и каталоги, которые вложены в переданную директорию. Функция рекурсивная. Если второй параметр задан в *true*, это значит что саму передаваемую папку необходимо оставить. Как правило, данный метод вызывается при деинсталляции модуля. В отличии от pub_dir_create_in_files, здесь необходимо указывать полный путь от корня сайта.

Пример:

```
function uninstall_children($id_module)
{
   global $kernel;

   $kernel->pub_dir_recurs_delete('content/files/'.$id_module');
}
```

Другие функции

pub_files_list_get

$kernel → pub_files_list_get ($path)

Назначение

Получить массив файлов в указанном каталоге.

Параметры

- $path: *String*
  Путь к папке, чьё содержимое необходимо узнать.

Возвращаемое значение

- *Array*

Описание

Возвращает массив файлов содержащихся в каталоге, переданном в качестве параметра. В качестве ключа элемента массива используется имя файла вместе с путём к нему, а в качестве значения – только имя файла.

Другие функции

pub_image_save

$kernel → pub_image_save($ufile, $id, $path_full_image, $big=0, $thumb=0, $watermark_image=0)

Назначение

Меняет размеры загруженной картинки до нужных размеров и сохраняет ее в каталог.

Параметры

- $ufile: *String*
  Путь и имя файла обрабатываемого изображения.
- $id:*Int*
  Начальная часть имени файла уже обработанного изображения, к которой будет добавлена уникальная составляющая.
- $path_full_image: *String*
  Путь, куда будет сохранено изменённое изображение.
- $big: *Array*
  Массив с параметрами для формирования БОЛЬШОГО изображения. Если 0, то данное изображение не формируется.
- $thumb: *Array*
  Массив с параметрами для формирования МАЛОГО изображения. Если 0, то данное изображение не формируется.
- $watermark_image: *Array*
  Массив с параметрами для формирования водяного знака на БОЛЬШОМ изображении.

Возвращаемое значение

- *String*

Описание

Метод обрабатывает изображения в трёх форматах: jpg, gif, png. Из исходного изображения могут быть сформированы большое и малое изображение, кроме того, к большому изображению может быть добавлена защита в виде «водяного знака».

В качестве параметра для формирования большого изображения передаётся массив следующего вида:

```
$big['width'] = 400;
$big['height'] = 300;
```

В массиве указываются значения длины и ширины, к которым должно быть приведено большое изображение.

Аналогичным образом указывается массив параметров для создания маленького изображения:

```
$big['width'] = 100;
$big['height'] = 75;
```

Следует учитывать, что ширина и высота обработанных изображений может отличаться, если будут нарушаться пропорции исходного изображения. При выполнении масштабирования предпочтение отдаётся сохранению пропорций, а затем ширине изображения.

Для добавления водяного знака необходимо определить массив его настроек и передать его в метод. Массив выглядит следующим образом:

```
$watermark_image[‘path’] = ‘content/files/fatermark.gif’;
$watermark_image[‘place’] = 0;
$watermark_image[‘transparency’] = 30;
```

Ключ *path* указывает путь и имя файла водяного изображения. Ключ *place* определяет местоположение водяной марки относительно большого изображения, и может принимать следующие значения:

- 0 – по центру;
- 1 – левый верхний уровень;
- 2 – правый верхний угол;
- 3 – правый нижний угол;
- 4 – левый нижний угол;

Последний ключ (*transparency*) определяет уровень прозрачности, в процентах, который должен иметь водяной знак. Принимает значения от 1 до 100 и если не задан, то равен 50.

После выполнения метода будет создано два (или одно) изображения, которые будут помещены в папку $path_full_image, при этом большое изображение непосредственно помещается в эту папку, а маленькое помещается во вложенную папку с именем ‘tn’ ($path_full_image.’/tn’). Имена файлов большого и маленького изображения будут одинаковы.

Пример:

```
//Путь к обрабатываемому файлу
$tmp_name = 'temp/temp.jpg'

//Параметры большого изображения
$big = array(
   'width' => 400,
   'height' => 300
);

//Параметры малого изображения
$thumb = array(
   'width' => 100,
   'height' => 75
);

//Параметры водяной марки
$watermark_image = array(
   ‘path’ => ‘content/files/fatermark.gif’,
   ‘place’ => 3,
   'transparency' => 25
);
```

```
//Задаём путь для сохранения обработанных изображений.
//такой путь должен существовать
$path_to_save = 'content/images/'.$kernel->pub_module_id_get();
$filename = $kernel->pub_image_save($tmp_name, 'img', $path_to_save, $big, $thumb, $watermark_image);
```

Если взять в качестве идентификатора модуля значение ‘news’, то будут созданы следующие файлы:

```
content/images/news/img_345222534.jpg  //большое изображение
content/images/news/tn/img_345222534.jpg //малое изображение
```

А переменная $filename будет содержать: *img_345222534.jpg*

Другие функции

pub_image_text_write

pub_image_text_write

$kernel → pub_image_text_write ($image, $text_arr, $output=false)

Назначение

Добавляет текст поверх изображения.

Параметры

- $image: *String*
  Путь к файлу с картинкой.
- $text_arr: *Array*
  Массив с параметрами добавляемого текста.
- $output: *String*
  Имя и путь файла для записи изменённого изображения. Если не задано, то картинка будет выведена на экран

Возвращаемое значение

- *Void | Image*

Описание

Метод предназначен для возможности включать в состав изображения произвольный текст.

Второй параметр определяет сам текст, добавляемый к изображению, но и положение этого текста относительно изображения, Данный массив выглядит следующим образом:

```
$text_arr[font] = '/content/files/fonts/tahoma.tff';//путь и имя файла шрифта, используемого для написания текста. Начинать с "/"
$text_arr[text] = 'Специальное предложение'; // непосредственно добавляемая фраза
$text_arr[x] = 10; //положение текста по ширине, относительно верхнего левого угла изображения
$text_arr[y] = 10; //положение текста по высоте, относительно верхнего левого угла изображения
$text_arr[color] = '#FF0000; //цвет текста в формате #RRGGBB
```

Если последний параметр не задан, то изменённое изображение будет выведено на экран, и

не будет сохранено.

Другие функции

pub_image_save

pub_site_root_get

$kernel → pub_site_root_get ()

Назначение

Возвращает путь к корню сайта.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Функция возвращает путь к public_html или аналогичной корневой директории, содержащей скрипты сайта. Путь выдается абсолютный, т.е. от "/" физической директории на сервере, где расположен сайт.

Другие функции

Сбор данных о посетителе

В этом разделе собраны все функции, которые позволяют получить информацию о текущем посетителе сайта (именно посетителе). Данная информация может обрабатываться модулем, при формировании своего контента или выполнении каких-то специфических действий.

Перечень всех методов

pub_tracker_enter_point_get	Возвращает первый URL, на который попал посетитель при входе на сайт.
pub_tracker_from_get	Возвращает URL, с которого посетитель попал на сайт.
pub_tracker_search_engine_get	Возвращает имя поисковой машины.
pub_tracker_search_word_get	Возвращает поисковое слово, по которому посетитель попал на страницу сайт.
pub_tracker_walking_path_get	Возвращает массив страниц, которые просматривал посетитель сайта.
pub_tracker_walking_time_get	Возвращает время, проведенное посетителем на сайте.

pub_tracker_enter_point_get

$kernel → pub_tracker_enter_point_get ()

Назначение

Возвращает первый URL, на который попал посетитель при входе на сайт.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Может использоваться любыми модулями для анализа или сбора данные о посетителе сайта.

Другие функции

pub_tracker_from_get

$kernel → pub_tracker_from_get ()

Назначение

Возвращает URL, с которого посетитель попал на сайт.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Возвращает URL, который просматривал посетитель, до того как перешёл на текущую страницу. Информация может использоваться любыми модулями для анализа или сбора данных о посетителях сайта.

Другие функции

pub_tracker_search_engine_get

$kernel → pub_tracker_search_engine_get ()

Назначение

Возвращает имя поисковой машины.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Возвращает название поисковой машины, через которую посетитель попал на сайт (мог попасть). Может использоваться любыми модулями для анализа или сбора данные о посетителе сайта.

Другие функции

pub_tracker_search_word_get

$kernel → pub_tracker_search_word_get ()

Назначение

Возвращает поисковое слово, по которому посетитель попал на страницу сайт.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Метод возвращает поисковый запрос, через который посетитель попал на сайт, конечно только в том случае, посетитель попал на сайт именно через поисковую систему. Может использоваться любыми модулями для анализа или сбора данные о посетителе сайта.

Другие функции

pub_tracker_walking_path_get

$kernel → pub_tracker_walking_path_get ()

Назначение

Возвращает массив страниц, которые просматривал посетитель сайта.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Может использоваться любыми модулями для анализа или сбора данные о посетителе сайта.

Другие функции

pub_tracker_walking_time_get

$kernel → pub_tracker_walking_time_get ()

Назначение

Возвращает время, проведенное посетителем на сайте.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Возвращает время (в секундах), которое посетитель сайта провёл на нём. Может использоваться любыми модулями для анализа или сбора данные о посетителе сайта.

Другие функции

Управление административным интерфейсом

В этом разделе собраны все функции, которые помогают модулю управлять своим административным интерфейсом. Ещё раз напомним, что для работы административного интерфейса, в модуле должны присутствовать две функции ***interface_get_menu()*** и ***start_admin()***

Через две эти функции осуществляется формирование управление административным интерфейсом. За более подробной информацией обращайтесь к документации по разработке модулей.

```
function interface_get_menu($menu)
{
      //Создаётся заголовок первого блока элементов меню
     $menu->set_menu_block('[#example_module_label_block_menu#]');
     $menu->set_menu("[#example_module_label_menu1#]","point1");
     $menu->set_menu("Второй пункт","point2");
     //Указываем ID меню по умолчанию
     $menu->set_menu_default('control_base');
  return true;
}

  function start_admin()
  {
     global $kernel;

     $html = '';
     //Прежде всего определяется пункт выбранного меню
     $select_menu = $kernel->pub_section_leftmenu_get();

     //В зависимости от выбранного пункта происходят разные действия
     switch ($select_menu)
     {
        case 'point1':
           $html = "Сформировали форму по управлению пользователями";
           break;
        case 'point2':
           $html = "Сформировали форму по управлению базой";
           break;
     }

     return $html;
  }
```

Перечень всех методов

pub_httpget_get	Возвращает массив $_GET, взятый из сессии.
pub_httpost_get	Возвращает массив $_POST, взятый из сессии.
pub_httppost_errore	Функция регистрация ошибки при обработке POST запроса.
pub_httppost_response	Подготовка ответа браузеру для форм, отправленных методом jspub_form_submit()
pub_module_id_get	Возвращает идентификатор текущего модуля.
pub_redirect_for_form	Возвращает ссылку для формы POST.
pub_redirect_refresh	Переход на заданный URL через JavaScript (перегрузка через Ajax).
pub_redirect_refresh_global	Глобальное перенаправление на заданный URL.
pub_redirect_refresh_reload	Переход на заданный URL, с указанием нижнего меню (перегрузка всей страницы).

pub_section_current_get	Возвращает идентификатор секции, в которой сейчас находится администратор сайта.
pub_section_leftmenu_get	Возвращает идентификатор левого меню, в котором сейчас находится администратор сайта.

pub_httpget_get

$kernel → pub_httpget_get ($name_var = "", $prepare = true)

Назначение

Возвращает массив $_GET, взятый из сессии.

Параметры

- $name_var: *String*
  Имя конкретной переменной, для получения её значения.
- $prepare: *Bool*
  Определяет, будет ли для переменной применён метод pub_str_prepare_set

Возвращаемое значение

- *Array* | *String* | *Int*

Описание

При каком либо обращении к сайту происходит запись массив $_GET и $_POST в сессию. С помощью этой функции модуль может получить доступ к данным, содержащимся в массиве $_GET.

В качестве параметра можно передать название переменной, которая должна находиться в этом массиве. Тогда будет возвращено сразу её непосредственное значение или пустая строка, если такой переменной нет.

Второй параметр по умолчанию задан в true. Он имеет смысл, если указано имя получаемой переменной, и это означает, что полученное значение будет обработано на предмет исключения SQL инъекций.

Другие функции

pub_httpost_get

pub_httpost_get

$kernel → pub_httppost_get ($name_var = "", $prepare = true)

Назначение

Возвращает массив $_POST, взятый из сессии.

Параметры

- $name_var: *String*
  Иимя конкретной переменной, для получения её значения.
- $prepare: *Bool*
  Определяет, будет ли для переменной применён метод pub_str_prepare_set

Возвращаемое значение

- *Array* | *String* | *Int*

Описание

При каком либо обращении к сайту происходит запись массив $_GET и $_POST в сессию. С помощью этой функции модуль может получить доступ к данным, содержащимся в массиве $_POST.

В качестве параметра можно передать название переменной, которая должна находиться в этом массиве. Тогда будет возвращено сразу её непосредственное значение или пустая строка, если такой переменной нет.

Второй параметр по умолчанию задан в true. Он имеет смысл, если указано имя получаемой переменной, и это означает, что полученное значение будет обработано на предмет исключения SQL инъекций. В случае необходимости, может потребоваться отключить автоматическую проверку.

Другие функции

pub_httppost_errore

$kernel → pub_httppost_errore ($message, $critical = false)

Назначение

Функция регистрация ошибки при обработке POST запроса.

Параметры

- $message: *String*
  Сообщение об ошибке.
- $critical: *Bool*
  Если True, то ошибка считается критической и выполнение останавливается.

Возвращаемое значение

- *Void* | *String*

Описание

Функция вызывается при выполнении обработки POST запроса в административном интерфейсе. При этом, данный пост запрос должен быть получен с помощью функции

jspub_form_submit()

Если данная ошибка критическая (то есть дальше обработка запроса должна быть прервана), то используется второй параметр (значение TRUE) в этом, случае будет возвращаться результат идентичный тому, что возвращает функция pub_httppost_response.

Пример использования:

```
public function start_admin()
{
   global $kernel;

   $content = '';
   switch ($kernel->pub_section_leftmenu_get())
   {
      case 'edit_show':
        // Вызов метода для построения формы редактирования
        // отправка этой формы осуществляется с использованием
        // функция jspub_form_submit()
           $content = $this->form_show();

        break;

      case 'edit_save':
        // Вызов метода для сохранения
           $content = $this->form_save();
         break;
    }

   return $content;
}

function form_save()
{
   global $kernel;
   //Забрали параметры из формы
   $name = $kernel->pub_httppost_get('name');
   if (empty($name))
   {
      //Форма заполнена не полностью, это критическая ошибка
     return $kernel->pub_httppost_errore('Форма заполнена не полностью', true);
   }
}
```

Другие функции

pub_httppost_response

pub_httppost_response

$kernel → pub_httppost_response ($message = '', $link_reload = '')

Назначение

Подготовка ответа браузеру для форм, отправленных методом jspub_form_submit()

Параметры

- $message: *String*
  Сообщение, выводимое в случае, если не было ошибок.
- $link_reload: *Bool*
  Идентификатор левого меню, на который необходимо осуществить переход.

Возвращаемое значение

- *String*

Описание

Функция обрабатывает (накапливает) сообщения об ошибках и формирует ответную строку для возврата её браузеру и обработки JavaScript-ом. В качестве параметра может быть передано сообщение, которое будет отображено администратору и автоматически скроется через несколько секунд. Кроме того можно задать идентификатор левого пункта меню, на который нужно перейти в случае отсутствия ошибок. Если такой идентификатор не задан, то форма останется без изменений.

Пример использования:

```
public function start_admin()
{
   global $kernel;

   $content = '';
   switch ($kernel->pub_section_leftmenu_get())
   {
      case 'edit_show':
        // Вызов метода для построения формы редактирования
        // отправка этой формы осуществляется с использованием
        // функция jspub_form_submit()
           $content = $this->form_show();

        break;

      case 'edit_save':
        // Вызов метода для сохранения
           $content = $this->form_save();
        break;
    }

   return $content;
}

function form_save()
{
   global $kernel;
   //Забрали параметры из формы
   $name = $kernel->pub_httppost_get('name');
```

```
    if (empty($name))
    {
      //Форма заполнена не полностью, это критическая ошибка
      return $kernel->pub_httppost_errore('Форма заполнена не полностью', true);
    }
}
```

Другие функции

pub_httppost_errore

pub_module_id_get

$kernel → pub_module_id_get($check_front_only = false)

Назначение

Возвращает идентификатор текущего модуля.

Параметры

- $check_front_only: *Bool*
  Если *true* - то обрабатывается только текущий модуль фронт-офиса

Возвращаемое значение

- *String*

Описание

Функция может использоваться как в публичных методах модуля, так и в административном интерфейсе модуля.

Параметр *$check_front_only* используется только ядром и другими системными функциями, в тех случаях, когда необходимо узнать идентификатор модуля, который обрабатывается именно при работе фронт-офиса.

Другие функции

pub_redirect_for_form

$kernel → pub_redirect_for_form ($url, $set_left_menu = true)

Назначение

Возвращает ссылку для формы POST.

Параметры

- $url: *String*
  Адрес, на который необходимо осуществить переход
- $set_left_menu: *Bool*
  Признак того, что URL начинается с идентификатора левого пункта меню.

Возвращаемое значение

- *String*

Описание

Формирует ссылку для использования её в поле "action" тега <*form*>. Переход осуществляется целиком во всём окне.

После указания пункта левого меню, могут идти дополнительные параметры, передаваемые через GET запрос.

Другие функции

pub_redirect_refresh

$kernel → pub_redirect_refresh ($url)

Назначение

Переход на заданный URL через JavaScript (перегрузка через Ajax).

Параметры

- $ url: *String*
  Адрес, на который необходимо осуществить переход.

Возвращаемое значение

- *String*
  Возвращается код для перехода.

Описание

Будет выведен код JavaScript для перезагрузки только области контента. Передаваемый параметр должен начинаться с идентификатора действия, которое будет доступно через метод pub_section_leftmenu_get. После этого могут идти дополнительные параметры.

Метод используется в административном интерфейсе, и следует помнить, что его результат должен быть возвращён (выведен в поток), с тем что бы произошел переход на указанный URL.

Пример:

```
function start_admin()
{
   $html = '';
   $action = $kernel-> pub_section_leftmenu_get();

   switch ($action)
   {
      case 'myaction_start':
         //выполняемые действия
```

```
        //Возвращаем код для перехода
        $html = $kernel-> pub_redirect_refresh (“myaction&param1=value1&param2=value2”);
        break;

      case ‘myaction’:
        //выполняемые действия
        $p1 = $kernel->pub_httpget_get(“param1”);
        $p2 = $kernel->pub_httpget_get(“param2”);
        $html = $p1.”-“.$p2;
        break;
    }
    return $html;
}
```

Другие функции

pub_redirect_refresh_global

$kernel → pub_redirect_refresh_global ($url, $scheme = SSL_CONNECTION)

Назначение

Глобальное перенаправление на заданный URL.

Параметры

- $url: *String*
  Адрес, на который необходимо осуществить переход.
- $scheme: *Bool*
  Признак перехода по защищенному протоколу.

Возвращаемое значение

- *Void*

Описание

В качестве параметра передаётся URL, начиная от доменного имени, на который необходимо осуществить перенаправление.

После применения этого метода в административном интерфейсе происходит переход на заданный адрес.

Другие функции

pub_redirect_refresh_reload

$kernel → pub_redirect_refresh_reload ($url, $scheme = SSL_CONNECTION)

Назначение

Переход на заданный URL, с указанием нижнего меню (перегрузка всей страницы).

Параметры

- $url: *String*
  Адрес, на который необходимо осуществить переход
- $scheme: *Bool*
  Признак перехода по защищенному протоколу.

Возвращаемое значение

- *Void*

Описание

Работа функции идентична работе pub_redirect_refresh, с той лишь разницей, что перегружается вся страница, а не только область конента.

Пример:

```
function start_admin()
{
   $html = '';
   $action = $kernel-> pub_section_leftmenu_get();

   switch ($action)
   {
      case 'myaction_start':
         //выполняемые действия

         //Сразу переход
         $kernel-> pub_redirect_refresh_reload ("myaction&param1=value1&param2=value2");
         break;

      case 'myaction':
         //выполняемые действия
         $p1 = $kernel->pub_httpget_get("param1");
         $p2 = $kernel->pub_httpget_get("param2");
         $html = $p1."-".$p2;
         break;
   }
   return $html;
}
```

Говоря ещё раз о разнице pub_redirect_refresh и этой функцией стоит отметить, что последняя производит полную перегрузку административного интерфейса.

Другие функции

pub_section_current_get

$kernel → pub_section_current_get ()

Назначение

Возвращает идентификатор секции, в которой сейчас находится администратор сайта.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

При использовании данной функции в административном интерфейсе, будет всегда возвращен идентификатор базового модуля в независимости от того, какой тип административного интерфейса использует модуль.

Данный метод удобно использовать для получения идентификатора базового модуля, если все дочерние модули используют одни базу данных, и необходимо получить часть имени этой базы данных.

Другие функции

pub_section_leftmenu_get

pub_section_leftmenu_get

$kernel → pub_section_leftmenu_get ()

Назначение

Возвращает идентификатор левого меню, в котором сейчас находится администратор сайта.

Параметры

Отсутствуют.

Возвращаемое значение

- *String*

Описание

Возвращает идентификатор пункта левого меню, который является текущим. Используя данный метод, административный интерфейс модуля определяет те, действия которые ему необходимо выполнить.

Пример использования:

```
/**
* Функция формирования меню модуля
*
* @param pub_interface $menu
*/
public function interface_get_menu($menu)
{
   $menu->set_menu_block('Заголовок блока');
```

```
    $menu->set_menu('Управление кодом', 'code_manage');
    $menu->set_menu('Загрузка кода', 'code_load');
    $menu->set_menu_default('code_manage');
}

/**
* Функция отображения административного интерфейса
*
* @return string
*/
public function start_admin()
{
    global $kernel;

    $content = '';
    switch ($kernel->pub_section_leftmenu_get())
    {
        case 'code_manage':
            // Необходимые действия
            break;

        case 'code_load':
            // Необходимые действия
            break;
    }

    return $content;
}
```

Другие функции

pub_section_current_get

Функции взаимодействия

В этом разделе собраны все функции, которые могу быть использованы как для управления и формирования административного раздела, так и для взаимодействия модуля и ядра CMS в публичных модулях.

Перечень всех методов

debug	Метод для вывода отладочной информации.
pub_mail	Производит отсылку электронных писем заданным адресатам.
pub_mapsite_cashe_create	Формирует массив с картой сайта.
pub_mapsite_get	Возвращает массив со всеми страницами сайта.
pub_modul_properties_get	Возвращает значения свойства модуля.
pub_module_serial_get	Возвращает массив со всеми дополнительными настройками модуля.
pub_module_serial_set	Записывает массив со всеми дополнительными настройки модуля.
pub_modules_get	Возвращает массив подключенных модулей.
pub_page_current_get	Возвращает идентификатор страницы сайта, на которой находится пользователь или администратор.
pub_page_property_get	Возвращает конкретное свойство конкретной страницы (с наследованием, если это нужно).
pub_page_textlabel_replace	Заменяет текстовые метки, в переданном контенте, на их языковое представление.
pub_page_title_add	Добавляет текст к строке, возвращаемой методом ядра "вернуть тайтл".
pub_prefix_get	Возвращает префикс для имён таблиц mySql.
pub_str_prepare_get	Функция для обработки строки, после взятия из mySql.
pub_str_prepare_set	Функция подготовки строки для записи в mySql.
pub_table_tr_class	Возвращает класс строки таблицы, для использования в HTML.
pub_template_parse	Разбирает шаблон на блоки и помещает в массив.
pub_waysite_get	Возвращает массив страниц, входящих в дорогу.
pub_waysite_set	Добавляет страницы в кэш дороги.
runSQL	Выполняет MySQL запрос к базе данных.

debug

$kernel → debug ($text, $direct_output = false)

Назначение

Метод для вывода отладочной информации.

Параметры

- $text: *String* | *Array* | *Int*
  Передаётся контент для вывода.
- $direct_output: *Bool*
  Если *true*, то вывод производится непосредственно в поток, в момент вызова. В противном случае отладочная информация выводиться после отработки всего кода.

Возвращаемое значение

- *Void* | *String*

Описание

Метод используется для вывода отладочной информации.

Другие функции

pub_mail

$kernel → pub_mail($toaddr, $toname, $fromaddr, $fromname, $subject, $message, $attach=false, $hostname="", $att_files="")

Назначение

Производит отсылку электронных писем заданным адресатам.

Параметры

- $toaddr: *Array*
  Массив адресов получателей письма.
- $toname: *Array*
  Имена получателей письма. Ключи должны соответствовать ключам в массиве $toaddr.
- $fromaddr: *String*
  Адрес отправителя письма.
- $fromname:*String*
  Имя отправителя письма.
- $subject: *String*
  Тема письма.
- $message: *String*
  Тело письма. Может содержать HTML.
- $attach: *Bool*
  Если *true*, то к телу письма будут прикреплены изображения, ссылки на которые встретились в теле письма.
- $hostname: *String*
  Адрес хоста, где находятся изображения, которые могут быть прикреплены. Если не задан, или равен "" (пустое значение), то используется имя хоста, на котором работает сайт.
- $att_files: *Array*
  Массив файлов, которые должны быть прикреплены к письму. На данный момент эта

возможность не реализована.

Возвращаемое значение

- *Int*
  Количество отосланных писем.

Описание

Метод позволяет осуществить отправку электронных писем с сервера, на котором размещён сайт. Письма могут быть отправлены сразу нескольким адресатам. Кроме, того к письму могут быть приложены изображения, ссылки на которые есть в самом письме.
Пример использования:

```
global $kernel;
//Имя и адрес получателя
$toname[0] = "Сергей Петров";
$toaddr[0] = "sergey.p@mymail.ru";

//Имя и адрес отправителя
$fromname = "Робот с сервера ".$_SERVER['HTTP_HOST'];
$fromaddr = "noreply@".$_SERVER['HTTP_HOST'];

//Заголовок сообщения
$subject = "Автоматическое письмо с сайта ".$_SERVER['HTTP_HOST'];

//Текст сообщения
$message = "Hello <b>word</b>!";

//Отправка сообщения
$kernel->pub_mail($toaddr, $toname, $fromaddr, $fromname, $subject, $message);
```

Другие функции

pub_mapsite_cashe_create

$kernel → pub_mapsite_cashe_create ($type = 0, $id ="")

Назначение

Формирует массив с картой сайта.

Параметры

- $type: *Int*
  Тип выходного массива. 0 - линейный массив. 1 = древовидный массив.
- $id: *String*
  Идентификатор страницы, от которой формируется структура. По умолчанию формируется структура всех страниц сайта.

Возвращаемое значение

- *Array*

Описание

Функция используются модулями при необходимости получить структуру сайта.

Возвращаемый массив может быть линейным (для удобства обработки) или древовидным.

Возвращаемый массив имеет следующий вид:

```
[идентификатор страницы] =>
                          [“caption”]    => String //
                          [“properties”] => Array
                          [“curent”]     => Bool //флаг того, что данная страница является
текущей
                          [“include”]    => Array // массив страниц, подчинённых данной.
```

Другие функции

pub_mapsite_get

pub_mapsite_get

$kernel → pub_mapsite_get ()

Назначение

Возвращает массив со всеми страницами сайта.

Параметры

Отсутствуют.

Возвращаемое значение

- *Array*

Описание

Возвращаемый массив имеет линейную структуру. В качестве ключа используется идентификатор страницы. В качестве значения - другой массив со всеми свойствами страницы, в том числе и свойствами, добавленными модулями.

Результаты метода можно использовать для отображения структуры сайта в том или ином виде. Не рекомендуется его использовать для анализа свойств страниц.

Возвращаемый массив имеет следующий вид:

```
[index] => Array
    (
  [id] => index //Дублируется ID страницы
  [parent_id] =>                //ID родительской страницы (если есть)
  [caption] => Главная страница     //Название страницы
  [curent] => 1                 //Признак того что страница является текущей (true, false)
  [properties] => Array             //Массив дополнительных свойств
    (
      [title_other] => 1        //Признак того что есть title
      [name_title] => Наша компания //Непосредственно title страницы
      [template] => design/in.html  //Используемый шаблон
      [link_other_page] => about    //ID другой страница сайта,
                            //на которую осуществляется перенаправление пользователя
```

```
        )
    )
```

Кроме этого, в *[properties]* могут содержаться значения свойств модулей к странице, если в системе есть инсталлированные модули, прописавшие эти свойства, и значения свойства задано у конкретной страницы. В этом случае ключом для обращения к данному свойству будет *[<IDмодуля>_<IDсвойства>]*

Все значения берутся без учета наследования, кроме того, для любой странице в массиве *[properties]* всегда будут присутствовать ключи *[title_other]* и *[name_title]*. Остальные ключи в этом подмассиве могут отсутствовать, если их конкретные значения не указаны у конкретной страницы.

Другие функции

pub_mapsite_cashe_create

pub_modul_properties_get

$kernel → pub_modul_properties_get($name_prop, $id_modul = '', $nasled = true)

Назначение

Возвращает значения свойства модуля.

Параметры

- $name_prop: *String*
  Имя получаемого значения свойства.
- $id_modul: *String*
  Если необходимо, идентификатор модуля. Если значение не будет указано, то значение свойства будет искаться у модуля, из которого происходит вызов данного метода.
- $nasled*: Bool*
  Если *true*, то будет применено наследование, при получении значения (значение по умолчанию – *true*).

Возвращаемое значение

- *Array*

Описание

Функция может вызываться как для параметров текущего модуля (из которого происходит её вызов), так и с указанием идентификатора конкретного модуля, чей параметр необходимо узнать.

Помимо этого возможно указать признак, разрешающий получить значение параметра с использованием механизма наследования.

Для использования данной функции внутри конкретного модуля для получения значения свойства этого модуля необходимо указать только имя этого свойства

Пример использования:

```
$arr = $kernel->pub_modul_properties_get('user_template')
$arr['isset'];    // Признак того существует данное свойство или нет;
```

```
$arr['naslednoe'];  //Признак того унаследовано значение или нет
$arr['value'];     //Непосредственно значение
```

Другие функции

pub_module_serial_get

$kernel → pub_module_serial_get ($id = "")

Назначение

Возвращает массив со всеми дополнительными настройками модуля.

Параметры

- $id: *String*
  Идентификатор модуля, чьи параметры запрашиваются.

Возвращаемое значение

- *Array*

Описание

В данный массив модуль может записать любые значения, которые ему Необходимо сохранить. Пользователь никак не может влиять на эти значения, они доступны только с помощью функций ядра pub_module_serial_get() и pub_module_serial_set()

Если модуль обращается к своим данным, то указывать параметр функции не нужно.

Другие функции

pub_module_serial_set

pub_module_serial_set

$kernel → pub_module_serial_set ($array, $id = "")

Назначение

Записывает массив со всеми дополнительными настройки модуля.

Параметры

- $array: *String*
  Записываемый массив.
- $id: *String*
  Идентификатор модуля, чьи параметры сохраняются.

Возвращаемое значение

- *Array*:

Описание

В данный массив модуль может записать любые значения, которые ему Необходимо сохранить. Пользователь никак не может влиять на эти значения, они доступны только с помощью функций ядра pub_module_serial_get() и pub_module_serial_set()

Если модуль обращается к своим данным, то указывать параметр функции не нужно.

Другие функции
pub_module_serial_get

pub_modules_get

$kernel → pub_modules_get($id_modul = "")

Назначение
Возвращает массив подключенных модулей.

Параметры
- $id_modul: *Bool*
  Идентификатор базового модуля, для которого нужно выбрать дочерние.

Возвращаемое значение
- *Array*

Описание
При указании идентификатора базового модуля возвращается массив всех дочерних модулей, этого базового модуля. Если параметр не указан, то будет возвращён массив всех дочерних модулей, всех базовых модулей, проинсталлированных в CMS.

Другие функции

pub_page_current_get

$kernel → pub_page_current_get ()

Назначение
Возвращает идентификатор страницы сайта, на которой находится пользователь или администратор.

Параметры
Отсутствуют.

Возвращаемое значение
- String

Описание
Чаще всего используются в публичных методах модуля, но может быть применена и в административном интерфейсе модуля.

Если вызов произошел из публичного метода модуля, то возвращается идентификатор страницы сайта, которая формируются для пользователя. Полученное значение может быть использовано для формирования ссылок на эту страницу.

В случае вызова из административного интерфейса модуля, будет возращен идентификатор страницы, с которой работал (в последний раз) администратор сайта.

Другие функции
pub_module_id_get

pub_page_property_get

$kernel → pub_page_property_get($id_page, $id_prop, $nasledovat = true)

Назначение

Возвращает конкретное свойство конкретной страницы (с наследованием, если это нужно).

Параметры

- $id: *String*
  Идентификатор страницы.
- $id_prop:*String*
  Идентификатор свойства.
- $nasledovat: *Bool*
  Если *true*, то будет применено наследование, при получении значения (значение по умолчанию – *true*).

Возвращаемое значение

- *Array*

Описание

Функция применяется в публичных метода модулях и позволяет модулю получить доступ к значению свойства страницы, из тех свойств, которые модуль установил.

Пример использования:

```
$arr = pub_page_property_get($kernel->pub_page_current_get(),'show_info');
$arr['isset'];     // Признак того существует данное свойство или нет;
$arr['naslednoe']; //Признак того унаследовано значение или нет
$arr['value'];     //Непосредственно значение
```

Другие функции

pub_page_textlabel_replace

$kernel → pub_page_textlabel_replace ($html)

Назначение

Заменяет текстовые метки, в переданном контенте, на их языковое представление.

Параметры

- $html: *String*
  Строка контента, который подвергается обработке.

Возвращаемое значение

- *String*
  Обработанный контент, с заменёнными языковыми переменными.

Описание

Метод производит поиск всех языковых меток (*[#<языковая_метка>#]*) в переданной строке и производит их замену на соответствующее представление для текущего языка. Если такой языковой переменной нет, то она перестаёт быть языковой переменной и

заменяется на строкой с её идентификатором.

Другие функции
pub_page_current_get

pub_page_title_add

$kernel → pub_page_title_add ($text)

Назначение
Добавляет текст к строке, возвращаемой методом ядра "вернуть тайтл".

Параметры
- $text: *String*
  Текст, который добавляется к тайтлу.

Возвращаемое значение
- *Void*

Описание
С помощью этой функции добавляется дополнительная информация к тайтлу страницы. Между уже существующем тайтлом и вновь добавляемым добавляется символ “-“.

Другие функции
pub_waysite_set

pub_prefix_get

$kernel → pub_prefix_get ()

Назначение
Возвращает префикс для имён таблиц mySql.

Параметры
Отсутствуют.

Возвращаемое значение
- *String*

Описание
Возвращаемый префикс добавляется к имени таблиц, которые использует модуль.

Пример использования:

```
global $kernel;

$query = “SELECT * FROM `”.$kernel->pub_prefix_get().”_mymodul_data”;
$resource = $kernel->runSQL($query);
```

Другие функции

pub_str_prepare_get

$kernel → pub_str_prepare_get ($str)

Назначение

Функция для обработки строки, после взятия из mySql.

Параметры

- $str: *String*
  Обрабатываемая строка.
- $html: *Bool*
  Признак применения метода htmlspecialchars() к возвращаемой строке.

Возвращаемое значение

- *String*

Описание

Функция производит обработку строки, после того как она взята из mySql.

Другие функции

pub_str_prepare_set

pub_str_prepare_set

$kernel → pub_str_prepare_set ($str)

Назначение

Функция подготовки строки для записи в mySql.

Параметры

- $str: *String*
  Обрабатываемая строка.

Возвращаемое значение

- *String*

Описание

Функция производит проверку на использование магических кавычек и необходимость их экранирования.

Другие функции

pub_str_prepare_get

pub_table_tr_class

$kernel → pub_table_tr_class ($num, $str_dop = '')

Назначение

Возвращает класс строки таблицы, для использования в HTML.

Параметры

- $num: *String*
  Номер строки выводимой строки.
- $str_dop: *String*
  Возможные дополнения, которые нужно дописать помимо класса.

Возвращаемое значение

- *String*

Описание

По переданному номеру строки определяется чётная это строка или нет и в соответствии с этим возвращается либо пустая строка, либо строка с именем класса.

Другие функции

pub_template_parse

$kernel → pub_template_parse ($filename, $createlevel = false)

Назначение

Разбирает шаблон на блоки и помещает в массив.

Параметры

- $filename: *String*
- Имя файла шаблона.
- $createlevel: *Bool*
  Если *true*, то при отсутствии уровни всё равно создаются и имеют нулевой индекс.

Возвращаемое значение

- *Array*

Описание

Разбирает шаблон стандартного вида и формирует массив, где в качестве ключа используется метка, а значения - HTML код. Например, из шаблона…

```
<!--@begin -->
<table cellpadding="3" cellspacing="1" border="0" bordercolor="black" width="100%">
        <tr class="table_content_shapka" bgcolor="#F05800">
                <th>[#shop_admin_users_stat_caption1#]</th>
                <th>[#shop_admin_users_stat_caption2#]</th>
                <th>[#shop_admin_users_stat_caption3#]</th>
        </tr>
<!-- @line -->
        <tr class="table_content_str" bgcolor="#FFF3EB">
                <td>%username%</td>
                <td>%usernum%</td>
                <td>%usersum%</td>
        </tr>
<!-- @end -->
        <tr class="table_content_shapka" bgcolor="#F05800">
                <th>[#shop_admin_users_stat_caption4#]</th>
                <th>%itognum%</th>
                <th>%itogsum%</th>
        </tr>
</table>
```

… получится массив следующего вида:

```
Array
(
   [begin] =>
```

```
<table cellpadding="3" cellspacing="1" border="0" bordercolor="black" width="100%">
        <tr class="table_content_shapka" bgcolor="#F05800">
                <th>[#shop_admin_users_stat_caption1#]</th>
                <th>[#shop_admin_users_stat_caption2#]</th>
                <th>[#shop_admin_users_stat_caption3#]</th>
        </tr>
  *
  [line] =>
        <tr class="table_content_str" bgcolor="#FFF3EB">
                <td>%username%</td>
                <td>%usernum%</td>
                <td>%usersum%</td>
        </tr>
  *
  [end] =>
        <tr class="table_content_shapka" bgcolor="#F05800">
                <th>[#shop_admin_users_stat_caption4#]</th>
                <th>%itognum%</th>
                <th>%itogsum%</th>
        </tr>
</table>
)
```

Начало блока задаётся как "*<!-- @Имя_блока -- >*". Блок будет продолжаться до конца файла, либо до объявления другого блока. Шаблон может быть много уровневым. Для этого внутри любого блока вы можете определить зарезервированный блок вида "*<!-- @@nextlevel -- >*" (обратите внимания, должно стоять две "собачки").

Указание этого подблока говорит о том, что ваш блок будет иметь несколько подуровней. В этом случае, выходной массив для такого блока будет состоять и массива, где в качестве ключа число (уровень вложенности) а в качестве значений – HTML

Пример многоуровневого шаблона:

```
<!--@begin -->
Начало
  *
<!-- @line -->
Линия уровня 0 <br>
  *
<!-- @@nextlevel -->
   Линия уровня 1 <br>
  *
<!-- @@nextlevel -->
     Линия уровня 2 <br>
  *
<!-- @end -->
Конец
  *
```

Получившийся массив будет иметь вид:

```
Array
(
  [begin] => Начало
```

```
  *
  [line] => Array
        (
          [0] => Линия уровня 0
          [1] => Линия уровня 1
          [2] => Линия уровня 2
        )
  [end] => Конец
)
```

Другие функции

pub_waysite_get

$kernel → pub_waysite_get ($id_page = '')

Назначение

Возвращает массив страниц, входящих в дорогу.

Параметры

- $id: *String*
  Идентификатор страницы, от которой формируется дорога. По умолчанию выводится вся дорога.

Возвращаемое значение

- *Array*

Описание

Дорога - ветка структуры, в которой находиться пользователь (от корневой страницы, до текущей). Возвращаемый массив имеет следующий вид:

```
[pagefaq] => Array
  (
    [id] => pagefaq - ID страницы
    [parent_id] => - rus родительская страница
    [caption] => Вопросы и ответы - название (caption) страницы
    [properties] => Array - свойства страницы
      (
        [title_other] => 1 - флаг, указывающий на то, что title страницы должен быть иным,
чем ее название
        [name_title] => Вопросы и ответы - title страницы
      )

    [curent] => 1 - флаг, указывающий на то, что данная страница является текущей
  )
```

Другие функции

pub_waysite_set

$kernel → pub_waysite_set ($adds_array)

Назначение

Добавляет страницы в кэш дороги.

Параметры

- $adds_array: *Array*
  Массив с параметрами страницы, которая добавляется к дороге.

Возвращаемое значение

- *Void*

Описание

Передаваемый массив состоит из следующих ключей:

$adds_array[url] - URL, куда приведет клик по элементу дороги

$adds_array[caption] - название (текст) элемента дороги

Метод используется в тех модулях, в которых данные ещё как-то структурированы и необходимо добавить элементы навигации, для более удобной навигации по этой структуре и для большей интеграции с имеющейся структурой сайта.

Другие функции

runSQL

$kernel → runSQL ($sql = "", $link = "")

Назначение

Выполняет MySQL запрос к базе данных.

Параметры

- $sql: *String*
  Строка с выполняем SQL запросом.
- $link: *Resource*
  Ссылка на подключения к базе данных.

Возвращаемое значение

- *Void*

Описание

Выполняет переданный в качестве параметра SQL запрос. Если второй параметр не указан, то выполнение запроса происходит в основной базе данных.

```
global $kernel;
$query = "SELECT * FROM `".$kernel->pub_prefix_get()."_mymodul_data";
$resource = $kernel->runSQL($query);
```

Если по каким-то причинам необходимо использовать другую базу данных, то ссылка на неё передаётся в качестве второго параметра.

Функция имеет встроенный механизм оповещения об ошибках в SQL запросах, которые отправляются на электронный адрес, указанный в глобальных настройках.

Кроме того, при составлении SQL запросов не забывайте использовать префикс таблиц, возвращаемый с помощью функции pub_prefix_get.

Другие функции

Функции обработки данных

Здесь собранны функции, которые предназначены для облегчения стоящих переезд модулем задач по обработке и подготовке тех или иных данных.

Перечень всех методов

pub_array_convert_form	Преобразует массив для использования в HTML формах Ext`a.
pub_array_convert_form_rec	Преобразует массив и все его вложенные массивы для использования в HTML формах Ext`a.
pub_array_key_2_value	Замена переменных вида %name% на значения переданного массива.
pub_array_keys_check_set	Дополняет массив обязательными ключами.
pub_data_to_string	Преобразует дату формата MySql в представление даты принятое в России.

pub_array_convert_form

$kernel → pub_array_convert_form ($arr)

Назначение

Преобразует массив для использования в HTML формах Ext`a.

Параметры

- $arr: *Array*
  Преобразуемый массив.

Возвращаемое значение

- *Array*

Описание

Передаваемый в функцию массив преобразуются к виду:

```
[["key","val"],["key","val"],["key","val"], ...]
```

Такой массив используется при создании выборных полей в формах, основанных на компоненте Ext`a:

```
new Ext.form.ComboBox({
    fieldLabel: '[#statist_all_label_date#]',
    hiddenName:'codepage',
    name: 'codepage',
    store: new Ext.data.SimpleStore({
        fields: ['code', 'caption'],
```

```
        data : Массив с данными
    }),
    valueField: 'code',
    displayField:'caption',
    typeAhead: true,
    mode: 'local',
    triggerAction: 'all',
    emptyText:' Выберете период...',
    selectOnFocus:true,
    width:150
})
```

Другие функции

pub_array_convert_form_rec

pub_array_convert_form_rec

$kernel → pub_array_convert_form_rec ($array)

Назначение

Преобразует массив и все его вложенные массивы для использования в HTML формах Ext`a.

Параметры

- $array: *String*
  Преобразуемый массив.

Возвращаемое значение

- *Array*

Описание

Метод идентичен методу pub_array_convert_form и отличается от него только тем, что обрабатывает вложенные массивы.

Другие функции

pub_array_convert_form

pub_array_key_2_value

$kernel → pub_array_key_2_value ($html, $array)

Назначение

Замена переменных вида %name% на значения переданного массива.

Параметры

- $html: *String*
  Обрабатываемый контент.
- $array: *Array*
  Массив меток и их значений.

Возвращаемое значение

- *String*

Описание

Метод обрабатывает переданную строку на предмет наличия в ней переменных, заключённых в одинарные проценты (%myvar%), заменяя эти переменные на значения, переданные во втором параметре.

Переменная %myvar% будет заменена на значение, которое расположено в массиве по ключу myvar.

```
$html = "Ваше имя: %name% <br>Ваш возраст: %age%";
$data = array("name" => "Евгений", "age" => "27")

$html = pub_array_key_2_value($html, $data);
```

В результате, функция вернёт следующий результат:

```
Ваше имя: Евгений
Ваш возраст 27
```

Другие функции

pub_array_convert_form

pub_array_keys_check_set

$kernel → pub_array_keys_check_set ($verify, $input)

Назначение

Дополняет массив обязательными ключами.

Параметры

- $verify: *String*
  Эталонной массив.
- $input: *Array*
  Обрабатываемый, проверяемый массив.

Возвращаемое значение

- *Array*

Описание

Метод объединяет массив $input и $verify таким образом, что все элементы $verify, которых нет в $input, попадают в последний ($input).

Метод используются при обработке форм, для гарантированного присутствия всех необходимых ключей в исходном массиве.

Пример:

```
global $kernel;

$in = array('name' => 'Alex', 'age'=> '27');

$verify = array();
$verify['age']     = 'Not select';
$verify['company'] = 'Not select';
$verify['text']    = '';
```

```
$in = $kernel->pub_array_keys_check_set($verify, $in);
```

В результате, массив $in будет содержать следующие данные

```
[name]    => [Alex]
[age]     => [27]
[company] => [Not select]
[text]    => []
```

Другие функции

pub_data_to_string

$kernel → pub_data_to_string ($data, $w = false)

Назначение

Преобразует дату формата MySql в представление даты принятое в России.

Параметры

- $data: *String*
  Дата в формате MySql.
- $w: *Bool*
  Признак необходимости вывода дня недели.

Возвращаемое значение

- *String*

Описание

В качестве основного параметра передаётся строка с датой в формате времени MySql (ГГГГ-ММ-ДД), после может идти время, которое данной функцией не учитывается.

Возвращается дата в виде "<понедельник>, 12 октября 2001 г.

Другие функции

Функции Java Script

В этом разделе вы найдёте перечень функций JavaScript, которыми вы можете (и должны) пользоваться при разработке административного интерфейса модуля.

Собранные здесь функции используются в шаблонах административного интерфейса модулей.

Перечень всех методов

jspub_click	Эмитирует клик по левому пункту меню.
jspub_confirm	Эмитирует клик по левому пункту меню (с подтверждением).
jspub_disabled_change	Включает или выключает элементы HTML элементы административного интерфейса.
jspub_form_submit	Отправка формы без перегрузки страницы.

jspub_click

jspub_click (lnk)

Назначение

Эмитирует клик по левому пункту меню.

Параметры

- lnk: *String*
  URL, на который осуществляем переход.

Возвращаемое значение

- *Void*

Описание

Функция используется для отправки любых GET (и только их) запросов модулю в качестве параметра используется URL для перехода, который должен начинаться с идентификатора пункта левого меню.

Эта основная функция, которая для формирования переходов по административному интерфейсу. Идентификатор, с которого начинается параметр функции, будет доступен через стандартную функцию ядра $kernel → pub_section_leftmenu().

Пример шаблона административного интерфейса модуля:

```
<!-- @table_body-->
    <tr>
      <td><input type="checkbox" name="items[%id%]" value="%id%"></td>
      <td>%number%</td>
      <td>%date%</td>
      <td><a href="#" onclick="jspub_click('action_edit&id=%id%')">%header%</a></td>
      <td align="center">%available%</td>
```

```
        <td align="center">%lenta%</td>
        <td align="center">%rss%</td>
        <td>%author%</td>
        <td><a href="#" onclick="jspub_click('action_edit&id=%id%')"><img
src="/admin/templates/default/images/icon_edit.gif"
alt="[#news_item_action_edit#]"/></a> <a href="#"
onclick="jspub_confirm('action_remove&id=%id%', '[#news_delete_confirm#]')""><img
src="/admin/templates/default/images/icon_delet.gif"
alt="[#news_item_action_remove#]"/></a></td>
    </tr>
```

Приведена часть шаблона, по формированию строк с новостями (модуль новостей). Управляющие функции модуля будет выглядеть следующим образом:

```
/**
* Функция формирования меню модуля
*
* @param pub_interface $menu
*/
public function interface_get_menu($menu)
{
  $menu->set_menu_block('Заголовок блока');
  $menu->set_menu('Список новостей', 'news_list');
  $menu->set_menu_default('news_list');
}

/**
* Функция отображения административного интерфейса
*
* @return string
*/
public function start_admin()
{
  global $kernel;

  $content = '';
  switch ($kernel->pub_section_leftmenu_get())
  {
    case 'news_list':
      $content = “Список новостей”
      // Формируется список новостей в которых используются функции jspub_click() и
      // jspub_confirm()
      break;

    case 'action_edit':
      // Сюда передаётся управление после клика в шаблоне
      break;

    case 'action_remove':
      // Сюда передаётся управление после клика в шаблоне, и только в том случае
      // если пользователь подтвердит выведенное ему сообщение.
      break;

  }
```

```
    return $content;
}
```

Другие функции

jspub_confirm

jspub_confirm

jspub_confirm (dialog_action, dialog_message)

Назначение

Эмитирует клик по левому пункту меню (с подтверждением).

Параметры

- dialog_action: *String*
  URL, на который осуществляем переход.
- dialog_message: *String*
  Сообщение, которое нужно подтвердить, прежде чем будет осуществлён переход.

Возвращаемое значение

- *Void*

Описание

Аналогична функции jspub_click, но перед тем как осуществить переход выводится сообщение и переход осуществиться только в том случае, если пользователь подтвердит это сообщение (то есть ответит "Yes").

Функцию удобно использовать для выполнения действий связанных с удалением каких-либо данных и требующих подтверждения, при их выполнение.

Другие функции

jspub_click

jspub_disabled_change

jspub_disabled_change (elem_check, arr_elem)

Назначение

Включает или выключает элементы HTML элементы административного интерфейса.

Параметры

- elem_check: *String* | *Object*
  Передаётся идентификатор чекбокса или сам объект Ext`а чекбокса
- arr_elem: *String*| *Array*
  Массив идентификаторов (или объектов Ext`а) которые нужно включить или выключить.

Возвращаемое значение

- *Void*

Описание

Функция производит выключение (или включение) заданных элементов в зависимости от того отмечена или не отмечена галочка, переданная в качестве первого параметра.

В качестве первого параметра может быть передан либо идентификатор стандартного checkbox`a, либо объект чекбокса, созданного с помощью компонент Ext`a.

Второй параметр может быть как массивом идентификаторов, так и строкой с идентификатором, или же объектом Ext`a или массивом таких объектов, а кроме того можно передать массив в котором будут содержаться и идентификаторы и объекты Ext`a.

В любом случае найденные элементы HTML формы буду выключены или включены в зависимости от того, отмечен чекбокс или нет, переданный в качестве первого параметра.

Другие функции

jspub_click

jspub_form_submit

jspub_form_submit (form, url)

Назначение

Отправка формы без перегрузки страницы.

Параметры

- form: *String*| *Object*
  Идентификатор формы или непосредственно объект формы Ext`a
- url: *String*
  URL, начинающийся с идентификатора левого пункта меню, на который будет осуществлён POST запрос.

Возвращаемое значение

- *Void*

Описание

Функция производит отправку формы, без перегрузки всей страницы. Получаемый ответ обрабатывается и в зависимости от него (ответа) производится либо вывод сообщений об ошибках, либо переход на указанный URL.

Фомирования ответа производиться с помощью функций ядра pub_httppost_errore и pub_httppost_response

Такой способ отправки форм административного интерфейса позволяет произвести всю проверку введённых данных только в PHP коде модуля, и избавляет программиста от разработки дополнительных функций JavaScripta.

Другие функции

pub_httppost_errore и pub_httppost_response

Функции формирования меню

В этом разделе вы найдёте все функции, которые отвечают за формирование левого меню модуля.
Все эти функции доступны только внутри предопределённой функции interface_get_menu/

```
public function interface_get_menu($menu)
{
   $menu->set_menu_block('Заголовок блока');
   $menu->set_menu('Список новостей', 'news_list');
   $menu->set_menu_default('news_list');
}
```

Конечный вид меню зависит от порядке вызова того или иного метода.

Перечень всех методов

set_menu	Объявляет пункт меню, в текущем блоке.
set_menu_block	Объявляет имя блока меню.
set_menu_default	Объявляет имя блока меню.
set_menu_plain	Добавляет к текущему блоку произвольный контент.
set_tree	Добавляет в текущий блок дерево элементов.

set_menu

$menu → set_menu ($name = '', $id = '', $array = null)

Назначение

Объявляет пункт меню, в текущем блоке.

Параметры

- $name: *String*
  Языковая переменная или непосредственное название пункта меню.
- $id: *String*
  Идентификатор пункта левого меню, доступный в дальнейшем через функцию pub_section_leftmenu_get.
- $array: *Array*
  Массив с дополнительными параметрами, которые необходимо добавить к GET запросу, выполняемому при клике по левому пункту меню. Ключ массив – название параметра, а значение массива – значение параметра.

Возвращаемое значение

- *Void*

Описание

Задаёт пункт левого меню в текущем блоке меню. Для всех пунктов меню, которые будут создаваться после объявления блока, будут относиться к этому блоку.

Другие функции

set_menu_block

$menu → set_menu_block ($name)

Назначение

Объявляет имя блока меню.

Параметры

- $name: *String*
  Языковая переменная или непосредственное название блока.

Возвращаемое значение

- *Void*

Описание

Все пункты меню, добавленные после вызова конкретного set_menu_block, будут отнесены именно к этому блоку меню.

То есть, если вам необходимо в меню несколько блоков, то необходимо сначала объявить первый блок меню, затем вызвать методы, добавляющие эти пункты меню, а затем объявлять новый блок меню и уже для него объявлять новые пункты меню.

Другие функции

set_menu_default

$menu → set_menu_default ($name)

Назначение

Объявляет имя блока меню.

Параметры

- $name: *String*
  Идентификатор пункта левого пункта меню

Возвращаемое значение

- *Void*

Описание

Объявляет пункт левого меню, который будет текущим при первом открытии административного интерфейса модуля. При дальнейших переходах, текущий пункт меню запоминается ядром и при повторном обращении будет открыт именно тот пункт меню, на котором последний раз был пользователь.

Другие функции

set_menu_plain

$menu → set_menu_plain ($content)

Назначение

Добавляет к текущему блоку произвольный контент.

Параметры

- $content: *String*
  Строка с произвольным HTML контентом, который добавляется к текущему блоку.

Возвращаемое значение

- *Void*

Описание

Когда в область левого меню необходимо добавить произвольный контент, необходимо объявить новый блок и затем воспользоваться данным методом. Добавляемый контент может использовать функцию jspub_click и jspub_confirm, для осуществления переходов и их дальнейшей обработки.

Следует учитывать, что идентификаторы левого меню, не объявленные через метод set_menu не буду запоминаться в качестве текущих элементов.

Другие функции

set_tree

$menu → set_tree ($tree)

Назначение

Добавляет в текущий блок дерево элементов.

Параметры

- $tree: *Object*
  Объект класс data_tree

Возвращаемое значение

- *Void*

Описание

Аналогично методу set_menu_plain вместо меню в блок помещается другой контент, а именно дерево произвольной структуры.

Прежде чем добавлять дерево необходимо создать экземпляр объекта set_menu_plain определить его параметры и уже затем добавлять его к меню с помощью данного метода.

Другие функции

Использование редактора контента

Административному интерфейсу модуля может понадобиться возможность использовать встроенный в SantaFox редактор контента. Для этого необходимо использовать объект, созданный из класса edit_content.

В административном интерфейсе модуля редактор контента вызывается в составе создаваемой модулем формы, ниже приведена часть шаблона, в которой описывается форма, одно из полей которой редактируется с помощью редактора контента:

```
<!-- @form →
<form action="%form_action%" method="POST" enctype="multipart/form-data">
<input type="hidden" name="values[id]" value="%id%"/>
<table border="0">
  <tbody>
    <tr>
      <td width="100" align="right">[#news_property_header_label#]</td>
      <td><input type="text" name="values[header]" value="%header%" style="width: 100%;"></td>
    </tr>
    <tr>
      <td width="100" align="right">[#news_property_description_short_label#]</td>
      <td><textarea name="values[description_short]" style="width: 100%;"
rows="4">%description_short%</textarea></td>
    </tr>
    <tr>
      <td colspan="3"> </td>
    </tr>
    <tr>
      <td colspan="3" height="300">%description_full%</td>
    </tr>
    <tr>
      <td colspan="3"> </td>
    </tr>
    <tr>
      <td align="right">[#news_property_author_label#]</td>
      <td><input type="text" name="values[author]" value="%author%" style="width: 100%"></td>
      <td rowspan="5" width="200" align="center">%image_block%</td>
    </tr>
  </tbody>
</table>
</form>
```

Переменная *%description_full%* будет заменена редактором контента, для чего необходимо выполнить следующий набор операторов:

```
$template = $kernel->pub_template_parse('item_form.html');
$html = $template['form'];

$editor = new edit_content();
$editor->set_edit_name('content_html');
$editor->set_simple_theme(true);
$editor->set_content('This is content for edit');

$html = str_replace('%description_full%', $editor->create(), $html);
```

В примере выше показана только часть кода, по формированию формы административного интерфейса, которая (часть кода) отвечает за построения поля ввода текста средствами редактора контента.

Отредактированный такой контент можно будет стандартными способами, через функций $kernel->pub_httppost_get().

Перечень всех методов

create	Создает HTML код для возможности редактирования контента средствами встроенного редактора.
set_content	Устанавливает контент, подлежащий редактированию.
set_edit_name	Устанавливает имя тега TEXTAREA в HTML форме.
set_file	Устанавливает в качестве источника контенета определенный файл.
set_full_form	Устанавливает в качестве формы вывода редактора полную HTML форму.
set_simple_theme	

create

$editor → create ()

Назначение

Создает HTML код для возможности редактирования контента средствами встроенного редактора.

Параметры

Отсутствует.

Возвращаемое значение

- *String*
  HTML код для вставки в форму.

Описание

Собирает все параметры класса и в соответствии с ним формирует HTML код, который встраивается в страницу и позволяет редактировать переданный контент средствами встроенного HTML редактора.

Другие функции

set_content

$editor → set_content ($html)

Назначение
Устанавливает контент, подлежащий редактированию.

Параметры
- $html: *String*
  HTML с контентом, который необходимо редактировать.

Возвращаемое значение
- *Void*

Описание
Используется для передачи контента подлежащего редактированию в тех случаях, когда он хранится не в отдельном файле, а каким либо другим образом, например в базе mySQL.

Другие функции

set_edit_name

$editor → set_edit_name ($value)

Назначение
Устанавливает имя тега TEXTAREA в HTML форме.

Параметры
- $value: *String*
  Имя, по которому в дальнейшем обращаться к отредактированному объекту.

Возвращаемое значение
- *Void*

Описание
В объекте TEXTAREA, с установленным именем и будет производится редактирование контента. В дальнейшем, в массиве, полученном с помощью функции $kernel→httppost_get(), можно будет обратиться к отредактированному контенту по этому имени. Если данный метод не используется, то область контента называется *content*.

Другие функции

set_file

$editor → set_file ($name)

Назначение
Устанавливает в качестве источника контенета определенный файл.

Параметры

- $name: *String*
  Имя файл с контентом.

Возвращаемое значение

- *Bool*

Описание

Устанавливает в качестве источника редактируемого контента определённый файл. Автоматического сохранения отредактированного контента в этот же файл не происходит. Если файла с таким именем нет, то он автоматически создается. Создание каталога автоматически не происходит.

Другие функции

set_full_form

$editor → set_full_form ($value = true)

Назначение

Устанавливает в качестве формы вывода редактора полную HTML форму.

Параметры

- $name: *Bool*
  Признак того использовать полную форму или нет. Модули используют не используют полную форму.

Возвращаемое значение

- *Void*

Описание

Под полной HTML формой подразумевается законченная страница пригодная для самостоятельного и полноценного отображения браузером. Данный метод необходимо использовать в том случае, если редактор контента будет открываться в отдельном и самостоятельном окне.

Сейчас такой способ открытия редактора контента использует только ядро.

Другие функции

set_simple_theme

$editor → set_simple_theme ($value = true)

Назначение

Устанавливает простую форму функциональных возможностей редактора контента.

Параметры

- $value: *Bool*

Включает или выключает упрощенную форму редактора контента.

Возвращаемое значение

- *Void*

Описание

Используется в тех случаях, когда необходимо дать только самые минимальные возможности при работе с редактором. Сюда входит лишь выделение текста и использование списков

Другие функции

Заключение

Надеемся, что данное руководство дало вам всю необходимую информацию для разработки собственных модулей, работающих под управлением SantaFox™.