Оглавление

[Код Oxwall 1](#_Toc377473062)

[Ядро 1](#_Toc377473063)

[Системные плагины 2](#_Toc377473064)

[Базовый плагин 2](#_Toc377473065)

[Панель администратора 2](#_Toc377473066)

[Отладка 2](#_Toc377473067)

[Системные компоненты 2](#_Toc377473068)

[Кэш в БД 2](#_Toc377473069)

[Кэширование запросов к БД 3](#_Toc377473070)

[Лог в БД 3](#_Toc377473071)

[Кэш в БД без тегов 3](#_Toc377473072)

[Утилитные классы 3](#_Toc377473073)

[События 3](#_Toc377473074)

[Разработка визуальной части 4](#_Toc377473075)

[Контроллер 5](#_Toc377473076)

[Компонент 5](#_Toc377473077)

[Декоратор 5](#_Toc377473078)

[Drag-And-Drop 5](#_Toc377473079)

[Основные компоненты сайта 5](#_Toc377473080)

[Некоторые элементы, используемые для создания визуальных представлений 5](#_Toc377473081)

## Код Oxwall

Код Oxwall состоит из следующих основных частей:

* Ядро
* Системные плагины
* Пользовательские плагины

### Ядро

Ядро располагается в папке ow\_core. Представляет собой ядро фреймворка.

Ядро реализует следующие функции: управление приложением, механизм кэша, работа с БД, MVC модель, роутинг url, система событий, конструктор форм, отображение html страниц, управление темами, управление плагинами, сессии, очередь рассылки емейл, лог.

Также ядро содержит интерфейсы для пользовательских классов:

* OW\_CacheService - Кэш
* OW\_Storage - хранилище файлов
* OW\_BillingProductAdapter – товары для продажи
* OW\_BillingAdapter – платежный шлюз

### Системные плагины

В Oxwall 2 системных плагина:

* Базовый плагин
* Панель администратора

Системные плагины располагаются в папке ow\_system\_plugins. Системные плагины совместно с ядром реализуют базовый функционал CMS.

### Базовый плагин

Плагин «Базовый плагин» находится в папке \ow\_system\_plugins\base.

Базовый плагин реализует функционал, который можно условно разбить на следующие группы:

* Системный: cron, лог, работа с БД, кэш.
* Представление: управление страницами, плагинами, меню, темами, мультиязычность.
* Управление пользователями: регистрация пользователей, авторизация, роли, права, профиль пользователя, приглашения, подтверждение почты.
* Прикладной: продажа товаров, комментарии, рейтинги, теги, лайки, отправка почты.

Плагин хранит данные в таблицах с префиксом ow\_base\_.

### Панель администратора

Плагин «Панель администратора» находится в папке \ow\_system\_plugins\admin.

Панель администратора предназначена для администратора и реализует визуальное представление для управления объектами базового плагина.

## Отладка

В \ow\_includes\config.php можно включить флаги:

1. OW\_DEBUG\_MODE – включение показа сообщений об ошибках и исключениях php.

Настройка реакции на ошибки и исключения производится модификацией \ow\_core\error\_manager.php.

По умолчанию, в режиме отладки в случае исключения выполнение прекращается.

1. OW\_DEV\_MODE – отключение кэша страниц и запросов в БД. Сильно замедляет время генерации страниц.
2. OW\_PROFILER\_ENABLE - включение отладочной панели.

## Системные компоненты

### Кэш в БД

Автоматически очищаемый кэш данных с поддержкой тегов в таблицах base\_cache и base\_cache\_tag.

OW::getCacheManager()->save(value, key, tags, cacheLifeTime);

OW::getCacheManager()->load(key);

OW::getCacheManager()->clean(tags); - сброс кэша с установленными тегами

### Кэширование запросов к БД

При выполнении любого запроса на получение данных из БД есть возможность указать время кэширования результата запроса и теги. Кэш удалится автоматически по истечении заданного времени или при явном вызове функции clean(tags).

findById(id, cacheLifeTime, tags); - запрос с кэшированием результата

OW::getCacheManager()->clean(tags); - сброс кэша

### Лог в БД

Запись лога в таблицу ow\_base\_log

$logger = OW::getLogger('type');

$logger->addEntry(‘message’, ‘key’);

$logger->writeLog();

### Кэш в БД без тегов

Автоматически очищаемый кэш данных в таблице ow\_base\_db\_cache.

OW::getCacheService()->set(key, value, lifeTimeSeconds);

OW::getCacheService()->get(key);

### Утилитные классы

UTIL\_Image – обработка картинок.

UTIL\_Profiler – замер времени выполнения.

## События

Возможность синхронно рассылать уведомления о событиях между плагинами. При отправке сообщения передаются данные, которые получатели могут прочитать или модифицировать.

Отправка:

$event = new OW\_Event('base\_add\_comment',…); // trigger event comment add

OW::getEventManager()->trigger($event);

Подписка на получение и обработка:

function blogs\_on\_notify( OW\_Event $event ){ обработка };

OW::getEventManager()->bind('base\_add\_comment', 'blogs\_on\_notify');

Используется для следующих задач:

1. Получение уведомлений и реакции на события

* ON\_USER\_REGISTER – пользователь зарегистрировался
* ON\_USER\_LOGIN – логин пользователя
* ON\_BEFORE\_USER\_REGISTER – регистрируется новый пользователь
* ON\_USER\_EDIT – пользователь отредактировал свой профиль
* ON\_USER\_EDIT\_BY\_ADMIN – администратор отредактировал данные пользователя
* base\_add\_comment – добавлен комментарий на сайте
* ON\_JOIN\_FORM\_RENDER – начало регистрации нового пользователя. Обработчик сообщения проверяет код приглашения и выбрасывает исключение в случае неверного кода.
* friends.add\_friend – отправлен запрос на добавление в друзья
* friends.request-accepted - принят запрос на добавление в друзья
* feed.after\_comment\_add – в ленте новостей добавлен комментарий к объекту
* feed.after\_like\_added – в ленте новостей добавлен лайк к объекту

1. Интеграция в элементы любого рода. Отправитель сообщения запрашивает данные у всех. Любой желающий подписывается на сообщение и добавляет/меняет данные

* base.preference\_menu\_items – страницы модуля «предпочтения»
* base.preference\_section\_label – разделы модуля «предпочтения»
* base.preference\_add\_form\_element - опции модуля «предпочтения»
* join.get\_captcha\_field – поле капчи в форме регистрации
* console.collect\_items – кнопки консоли
* console.load\_list – окно уведомлений в консоли
* console.ping – количество новых элементов кнопки консоли
* base.add\_main\_console\_item – пункты меню кнопки профиля в консоли
* notifications.send\_list – уведомления пользователю
* feed.on\_entity\_add – новый элемент в ленте новостей
* notifications.on\_item\_send – уведомление на почту
* notifications.collect\_actions – список почтовых уведомлений
* usercredits.on\_action\_collect – список действий для получения кредитов

1. Отправка команд плагинам
   * friends.send\_friend\_request – отправить запрос на добавление в друзья
   * feed.action – добавить событие в ленту новостей
   * notifications.add – добавить уведомления
   * usercredits.action\_update – пересчитать кредиты
   * usercredits.track\_action – списать кредиты за действие
   * usercredits.grant – начислить кредиты пользователю
2. Запрос информации у плагинов
   * usercredits.check\_balance – достаточно ли баланса кредитов для действия
   * usercredits.get\_balance – баланс кредитов пользователя
   * usercredits.error\_message – текст сообщения о недостаточном балансе

Лог событий отображается на отладочной панели.

Чтобы получить уведомление первым, надо указать приоритет: bind(…,…, 0). По умолчанию, приоритет – 1000.

Чтобы не дать возможность получить уведомление другим получателям, надо в обработчике вернуть false или вызвать $event->stopPropagation().

## Разработка визуальной части

В Oxwall 3 типа визуальных объектов:

* Контроллер
* Компонент
* Декоратор

Для построения визуальной части используется шаблонизатор [Smarty](http://ru.wikipedia.org/wiki/Smarty). Документация по Smarty <http://www.smarty.net/docsv2/ru/smarty.for.designers.tpl>

### Контроллер

Контроллер отвечает за формирование одной или нескольких страниц целиком.

В Oxwall 2 основных типа контроллеров:

* OW\_ActionController - контроллер страницы сайта
* ADMIN\_CTRL\_Abstract - контроллер страницы панели администратора

### Компонент

Компонент отвечает за формирование небольшого блока на страницах и обработку событий. Он самостоятельно формирует данные по переданным параметрам.

В Oxwall 2 основных типа компонентов:

* OW\_Component – простой компонент
* BASE\_CLASS\_Widget - виджет, который администратор или пользователь могут добавлять на страницы: главная, dashboard, профиль, страница группы.

Компонент может включать в себя другие компоненты и декораторы.

### Декоратор

Декоратор – это html файл, отображающий переданные данные. Декоратор является функцией Smarty и может находится в любом шаблоне, в т.ч. в другом декораторе. Декораторы упрощают код представления, убирают дублирование кода и обеспечивают единообразное отображение повторяющихся данных на разных страницах.

Например, декоратор avatar\_item используется для отображения аватара пользователя, user\_list\_item - для отображения аватара и имени пользователя.

### Drag-And-Drop

Для того, чтобы дать возможность пользователю менять расположение виджетов или других элементов на странице, используются классы BASE\_CTRL\_ComponentPanel, BASE\_CMP\_DragAndDropEntityPanel, BASE\_CMP\_DragAndDropIndex, ADMIN\_CTRL\_Components, ADMIN\_CMP\_DragAndDropAdminPanel.

### Основные компоненты сайта

* BASE\_CMP\_Console\* - консоль (верхняя панель управления)
* BASE\_CMP\_MainMenu – верхнее меню
* BASE\_CMP\_BottomMenu – нижнее меню

### Некоторые элементы, используемые для создания визуальных представлений

* BASE\_CMP\_ContentMenu – меню
* BASE\_CMP\_SortControl – контрол сортировки данных (подменю)
* BASE\_CMP\_Paging – постраничный вывод
* BASE\_CMP\_Breadcrumb – хлебные крошки
* Form – форма ввода данных
* OW::getFeedback()->info – Всплывающее сообщение
* OW.ajaxFloatBox – всплывающее окно с функционалом AJAX. Пример - \ow\_plugins\skeleton\controllers\floatbox.php.

Детальное описание со скриншотами здесь <http://docs.oxwall.org/design:interface>

<http://docs.oxwall.org/_media/design:origin-style-guide.psd>

Прикладные компоненты:

* BASE\_CMP\_Comments – отображение блока комментариев
* BASE\_CMP\_TagSearch - Поиск по тегам
* BASE\_CMP\_EntityTagCloud – облако тегов
* BASE\_CMP\_Rate – рейтинг объекта
* BASE\_CMP\_AvatarUserListSelect – окно выбора пользователей
* BASE\_CMP\_AvatarUserList, BASE\_CMP\_MiniAvatarUserList, BASE\_CMP\_Users, BASE\_CMP\_UserList – списки аватаров пользователей

OW::getNavigation()->addMenuItem – добавить пункт в главное меню