

ImageCMS WishList

Compatible with "[Module documentation standard](#)"

ImageCMS

Last updated May 24, 2013.

[About](#)

[Link's](#)

[Installation](#)

[Module configuration](#)

[Module Administration](#)

[Site Integration](#)

[Вывод кнопок добавления в Список Желания](#)

[Также после установки модуля будут доступны следующие возможности:](#)

[Use case](#)

[Use Case: #1](#)

[Use Case: #2](#)

[Use Case: #3](#)

[Use Case: #4](#)

[Use Case: #5](#)

[Use Case: #6](#)

[Use Case: #7](#)

[Use Case 1 - 7 в виде схемы](#)

[Development info.](#)

[Класс ParentWishlist](#)

[Properties](#)

[Public methods](#)

[API's](#)

[Models](#)

[FAQ](#)

About

Wishlist - модуль который позволяет пользователям магазина работать со Списками Желания. Список Желаний является списком товаров, в который попадают те наименования, который пользователь желает приобрести, но это действие откладывается по некоторым причинам. Пользователь, создавший список желаний также обладает возможностью публично заявить об интересе к определенным наименованиям, поделившись ссылкой через социальные сети либо электронную почту.

Link's

Installation

Для начала работы с модулем его нужно [установить](#).

Module configuration

В настройках модуля (Модули->Все модули->wishlist), можно указать ограничения для:

- длины имени списка желания;
- количества списков;
- количества товаров в списках;
- длины комментария;
- длины описания;
- ширины изображения пользователя;
- высоты изображения пользователя;
- размера изображения(байт).

Окно настроек Списка Желаний:

Настройка Списков Желания	
Максимальная длина имени Списка Желаний	<input type="text" value="254"/>
Максимальное количество Списков Желаний	<input type="text" value="10"/>
Максимальное количество товаров в Списках Желаний	<input type="text" value="100"/>
Максимальная длина комментария	<input type="text" value="500"/>
Максимальная длина описания	<input type="text" value="1000"/>
Максимальная ширина изображения пользователя	<input type="text" value="200"/>
Максимальная высота изображения пользователя	<input type="text" value="200"/>
Максимальный размер изображения (байт)	<input type="text" value="2000000"/>

Module Administration

В администрировании модулем (Модули -> Все модули -> wishlist) есть возможности:

- просмотреть и управлять информацией пользователей которые имеют Списки желаний;
 - редактировать им'я, день рождения, описание;
 - удалять/загружать картинку.
- просмотреть и управлять их списками;
 - удалять;
 - редактировать.
- просмотреть и управлять наименованиями в списках;
 - удалять;
 - перемещать между списками;
 - редактировать комментарии;
- создавать новые списки.

Страница пользователей:


На “Странице пользователей” вводится список пользователей которые создали списки желаний. Наводится их имя, количество списков и количество товаров в списках.

<input type="checkbox"/>	ID	ФИО	Количество списков	Количество товаров в списках
<input type="checkbox"/>	400	Пользователь1	0	0
<input type="checkbox"/>	47	Пользователь2	2	3

Управления информацией пользователя:

На этой странице можно редактировать информацию о пользователе списка желаний.

Пользователь



Имя

День рождения:

Описание:

Страница создания списка пользователя:

Создать список

Название списка:

Создать список

Страница управления списками пользователя:

На этой странице наедается возможность управлять списками пользователя: удалять, редактировать, перемещать наименования в списках и удалять их.

Списки пользователя: Пользователь2

[← Вернуться](#) [Настройки](#)

Пользователь Списки Создать список

Список1

Тип списка: shared

Удалить

Редактировать

№	Отписаться	Товар	Комментарий
1	Удалить Переместить	JVC - CHX1500RF - FM Modulation	
2	Удалить Переместить	Pyle PLSQ10D Red Label Square	

Мой список

Тип списка: shared

Удалить

Редактировать

№	Отписаться	Товар	Комментарий
1	Удалить Переместить	Pyle PLT-AB8 Subwoofer - PLTAB8	

Страница редактирования списка:

Эта страница предназначена для редактирования одново списка желания. Здесь доступны возможности по изменению всего находящегося в списке: редактировать имя и тип списка, а также удалять наименования и изменять их комментарии.

Редактировать список:

[← Вернуться](#) [Настройки](#)

Список

Имя:

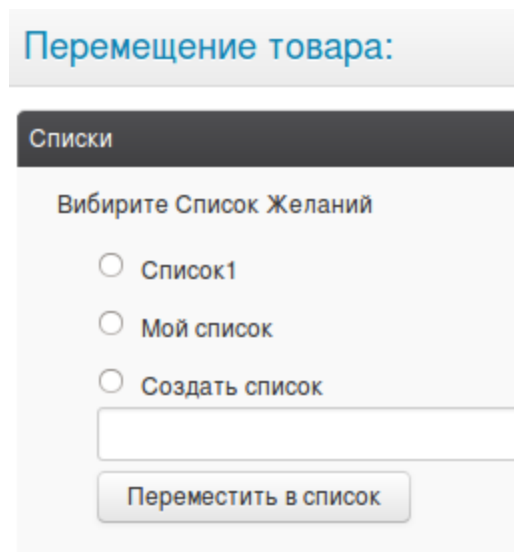
Тип списка:

№	Отписаться	Товар	Комментарий
1	Удалить	JVC - CHX1500RF - FM Modulation	<input type="text" value=""/>
2	Удалить	Pyle PLSQ10D Red Label Square	<input type="text" value=""/>

Сохранить

Страница перемещения наименования:

Страница предназначена для перемещения наименования в другой существующий или с возможностью создать новый список желания.



Site Integration

Вывод кнопок добавления в Список Желания

Для того чтоб пользователь мог добавить товар в Список Желания средствами модуля нужно добавить в шаблон строку:

```
{ $CI->load->module('wishlist')->renderWLButton(ИД_товара) }
```

Также после установки модуля будут доступны следующие возможности:

1. Перейдя по ссылке `site.url/wishlist` пользователю будут доступны просмотр всех Списков Желания, редактирование данных, создание нового списка;

upload

Удалить картинку

admin

23.01.1989 x [v] ▼

описание

Надіслати


Создать новый список

первый

Тип списка: **shared**

удалить

редактировать

Check	№	Отписатся	Товар	Коментарий
<input type="checkbox"/>	1	удалить	 Монитор PHILIPS 21.5" 222E2SB.00 16:9 DVI Black	
<p>Удалить</p>				

- Нажав на кнопку “Редактировать” - вы попадаете на страницу редактирования Списка Желания;

первый

public ▼

удалить

№	Отписатся	Товар	Коментарий
1	удалить	Монитор PHILIPS 21.5" 222E2SB.00 16:9 DVI Black	<div>коментарий</div>

Надіслати

- На странице `site.url/wishlist/all` - доступен список всех *публичных* Списков Желания. Используя его пользователи смогут просмотреть что хотят приобрести другие пользователи магазина.

Информация о пользователе

Имя: [admin](#)

Списки пользователя

[первый](#)



Дата рождения: 1989-01-23

Use case

Use Case: #1

Прецедент	Прецедент WL 1
Действующие лица	Пользователь Система
Цель	Добавить товар в WL для не авторизованного пользователя создавая новый WL.
Краткое описание	Войти в систему и добавить товар создавая новый WL

Действия исполнителя	Отклик системы
Не авторизованный пользователь добавляет товар в WL.	Система выдает окно авторизации пользователя.
Пользователь авторизуется.	Система возвращается к странице просмотра или страницы товара.
Пользователь добавляет товар в WL.	Система дает возможность создать новый WL или добавить товар в созданный WL.
Пользователь создает новый WL называя его в поле для введения названия.	Система ожидает.
Пользователь нажимает добавить WL.	Система создает WL с товаром в нем.

Use Case: #2

Прецедент	Прецедент WL 2
Действующие лица	Пользователь Система
Цель	Добавить товар у WL для не авторизованного пользователя используя существующий ВЛ.
Краткое описание	Войти в систему и добавить товар используя WL который уже созданный

Действия исполнителя	Отклик системы
Не авторизованный пользователь добавляет товар в WL.	Система выдает окно авторизации пользователя.
Пользователь авторизуется.	Система возвращается к странице просмотра или страницы товара.
Пользователь добавляет товар в WL.	Система дает возможность создать новый WL или добавить товар в созданный WL
Пользователь добавляет товар в существующий WL выбирая его с помощью кнопки.	Система ожидает.

Use Case: #3

Прецедент	Прецедент WL 2
Действующие лица	Пользователь Система
Цель	Добавить описание к существующему WL
Краткое описание	Описать WL который уже созданный

Действия исполнителя	Отклик системы
Пользователь переходит в список WL.	Система предоставляет список WL
Пользователь нажимает редактирования выбранного WL	Система дает доступ к избранному WL и инструменты управления им.

Пользователь вводит описание в поле данном для этого.	Система ожидает.
Пользователь нажимает сохранить изменения.	Система сохраняет изменения.

Use Case: #4

Прецедент	Прецедент WL 4
Действующие лица	Пользователь Система
Цель	Создать WL.
Краткое описание	Создать новый WL используя инструменты

Действия исполнителя	Отклик системы
Пользователь переходит в список WL.	Система предоставляет список WL и инструменты управления WL.
Пользователь нажимает кнопку «Создать WL» и заполняет поле имя.	Система ожидает.
Пользователь нажимает сохранить изменения.	Система создает новый WL.

Use Case: #5

Прецедент	Прецедент WL 5
Действующие лица	Пользователь Система
Цель	Переместить товар с одного WL к другому.
Краткое описание	Используя инструменты WL переместить товар.

Действия исполнителя	Отклик системы
Пользователь переходит в список WL.	Система предоставляет список WL и инструменты управления WL.

Пользователь выбирает товар в любом WL и нажимает на контекстное меню перемещения.	Система выдает список всех созданных WL.
Пользователь выбирает WL	Система переносит выбранный товар в выбранный WL.

Use Case: #6

Прецедент	Прецедент WL 6
Действующие лица	Пользователь Система
Цель	Добавить информацию о пользователе.
Краткое описание	Добавить информацию о пользователе которую смогут увидеть другие пользователи.

Действия исполнителя	Отклик системы
Пользователь переходит в список WL.	Система предоставляет список WL и инструменты управления WL.
Пользователь нажимает на добавлении аватару.	Система выдает дополнительное меню по загрузке аватара.
Пользователь выбирает файл загрузки и нажимает "Выбрать".	Система грузит аватар.
Система вставляет аватар в выделенной зоне для аватара.	Система ожидает.
Пользователь вводит свою дату рождения.	Система ожидает.
Пользователь вводит свое имя и фамилию.	Система ожидает.
Пользователь вводит описание в выделенном для этого месте.	Система ожидает.
Пользователь нажимает сохранить изменения.	Система сохраняет изменения введены пользователем.

Use Case: #7

Прецедент	Прецедент WL 7
-----------	----------------

Действующи е лица	Пользователь Система
Цель	Изменение вида действующего либо не действенного WL
Краткое описание	Изменение вида действующего либо не действенного WL с помощью инструментов

Действия исполнителя	Отклик системы
Пользователь переходит в список WL.	Система предоставляет список WL и инструменты управления WL.
Пользователь нажимает редактирования выбранного WL.	Система дает доступ к избранному WL и инструменты управления им.
Пользователь меняет тип с типа который указан по-стандарта на тип желаемый.	Система ожидает.
Пользователь нажимает сохранить.	Система хранит тип WL меняя его данные.

Use Case 1 - 7 в виде схемы

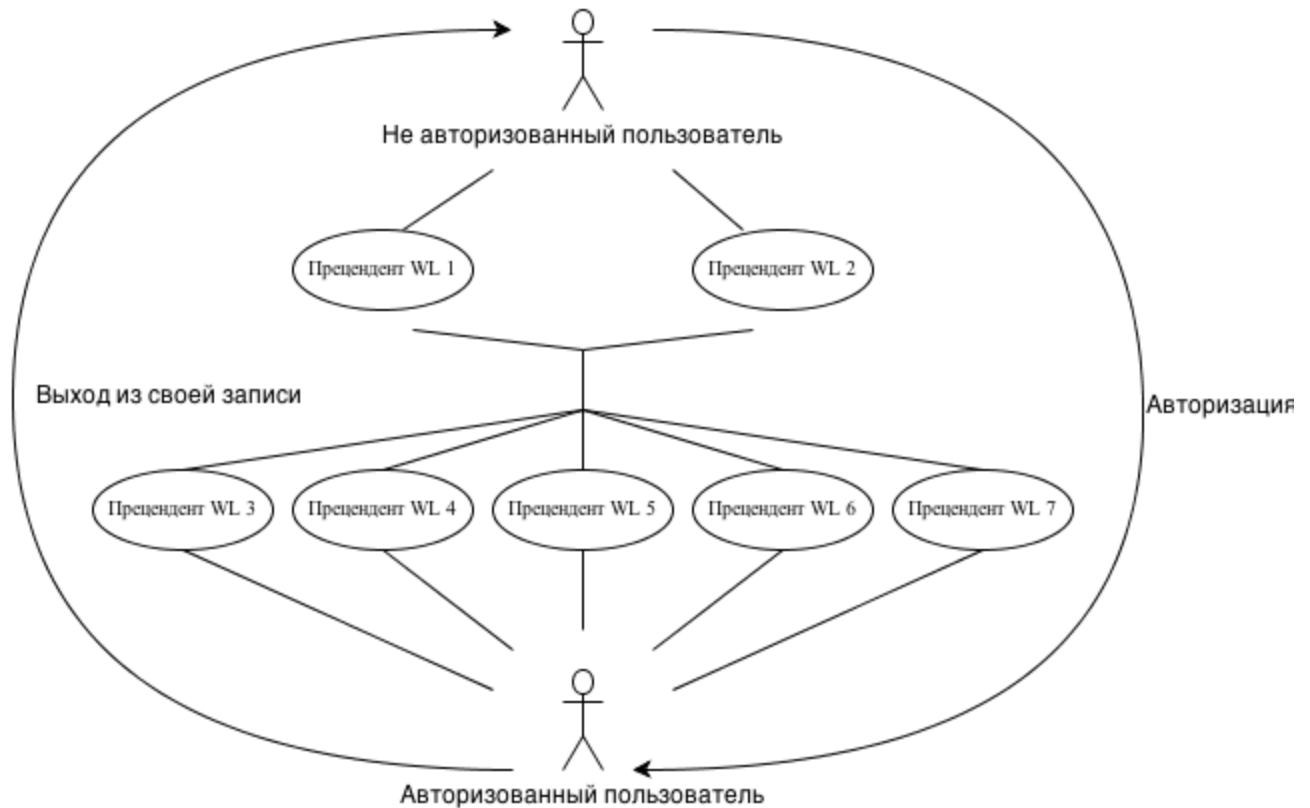


рис.1

Development info.

Класс ParentWishlist

Properties

- **\$settings** - массив, служит для хранения настроек модуля;
- **\$dataModel** - массив, служит для хранения результатов методов класса;
- **\$errors** - массив, служит для хранения ошибок методов класса.

Public methods

- **all()**
(array \$access = array('public'))
метод для получения всех публичных списков пользователей;
- **show()**
(string \$hash, array \$access = array('public'))

метод для получения публичного списка пользователя;

- **getMostViewedWishLists()**
(int \$limit = 10)
метод для получения наиболее просматриваемых списков;
- **user()**
(int \$user_id, array \$access = array('public'))
метод для получения публичных списков пользователя;
- **userUpdate()**
(int \$userID, string \$user_name, string \$user_birthday, string \$description)
метод для обновления информации о пользователе;
- **updateWL()**
(int \$id, array \$data, array \$comments)
метод для обновления списка;
- **createWishList()**
(int \$user_id, string \$listName)
метод для создания списка;
- **deleteWL()**
(int \$id)
метод для удаления списка;
- **deleteAllWL()**
(int \$UserID)
метод для удаления всех списков пользователя;
- **_addItem()**
(int \$varId, int \$listId, string \$listName, int \$userId = null)
метод для добавления наименования в список;
- **moveItem()**
(int \$varId, int \$wish_list_id, int \$to_listId = "", int \$to_listName = "", int \$user_id = null)
метод для перемещения наименования;
- **deleteItem()**
(int \$variant_id, int \$wish_list_id)
метод для удаления наименования;
- **getUserInfo()**
(int \$id)
возвращает информацию про пользователя;
- **getUserWL()**
(int \$userId, array \$access = array('public', 'private', 'shared'))
метод служит для возвращения списка пользователя;
- **renderUserWLEdit()**
(int \$wish_list_id, int \$userId = null)
метод для возвращения списков для редактирования;
- **do_upload()**

- (int \$userID = null)
 метод для загрузки изображения пользователя;
- **getMostPopularItems()**
 (int \$limit = 10)
 метод служит для возвращения наименований которых наибольшее количество в списках;
- **getUserWishListItemsCount()**
 (int \$user_id)
 метод возвращает количеств наименований в списках пользователя;
- **deleteItemsByIds()**
 (array \$ids)
 метод служит для удаления наименований по их идентификатору;
- **deleteImage()**
 (string \$image)
 метод для удаления изображения пользователя.

API's

Models

- **getSettings()**
 возвращает массив настроек модуля;
- **setSettings()**
 (array \$settings)
 проводит сериализацию и сохраняет полученный массив настроек;
- **getWishLists()**
 (int \$userID = NULL)
 получает массив Списков Желаний по ИД пользователя;
- **getAllUsers()**
 получает список всех пользователей которые создавал Списки Желания;
- **createWishList()**
 (int \$listName, int \$user_id, string \$access = 'shared')
 создает Список Желания;
- **addItem()**
 (int \$varId, int \$listId, string \$listName, int \$user_id = null)
 добавляет товар в Список Желания;

- **delWishListById()**
(int \$id)
удаляет Список Желания;
- **deleteItem()**
(int \$variant_id, int \$wish_list_id)
удаляет товар со Списка Желания;