DrupalEshop

Drupal7 - 7.8

version :1.0.0

Drupal is a general-purpose server-side scripting language originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications.PHP can be deployed on most web servers and as a standalone interpreter, on almost every operating system and platform free of charge.

COREMODULE

Chaos tool suite

This suite is primarily a set of APIs and tools to improve the developer experience. It also contains a module called the Page Manager whose job is to manage pages. In particular it manages panel pages, but as it grows it will be able to manage far more than just Panels..A library of helpful tools by Merlin of Chaos.For the moment, it includes the following tools:

Plugins -- tools to make it easy for modules to let other modules implement plugins from .inc files.
Exportables -- tools to make it easier for modules to have objects that live in database or live in code, such as 'default views'.
AJAX responder -- tools to make it easier for the server to handle AJAX requests and tell the client what to do with them.
Form tools -- tools to make it easier for forms to deal with AJAX.
Object caching -- tool to make it easier to edit an object across multiple page requests and cache the editing work.
Contexts -- the notion of wrapping objects in a unified wrapper and providing an API to create and accept these contexts as input.
Modal dialog -- tool to make it simple to put a form in a modal dialog.
Dependent -- a simple form widget to make form items appear and disappear based upon the selections in another item.
Content -- pluggable content types used as panes in Panels and other modules like Dashboard.
Form wizard -- an API to make multi-step forms much easier.
CSS tools -- tools to cache and sanitize CSS easily to make user-input CSS safe.

Entity API

It enables modules to work with any entity type and to provide entities.

This module extends the entity API of Drupal core in order to provide a unified way to deal with entities and their properties. Additionally, it provides an entity CRUD controller, which helps simplifying the creation of new entity types.

Features

This is an API module, so it doesn't provide any end-user features. However, users may enable the entity tokens module, which provides token replacements for all entity properties that have no tokens and are known to the entity API.

Overview

The module provides API functions allowing modules to create, save, delete, view or to determine access for any entity, i.e. entity_create(), entity_save(), entity_delete(), entity_view() and entity_access().

The entity API introduces a unique place for metadata about entity relationships and entity properties: hook_entity_property_info(). This information about entity properties contains the data type and callbacks for how to get and set the data of a property. Modules may rely on this information in order to support any entity property, e.g. Rules and the Search API build upon that.

Furthermore the module provides data wrappers that make use of the available information to provide a simple and unified access to entities and their properties. For usage examples have a look at the README or the provided tests.

Beside that, the module helps you defining a new entity type. For that, it provides an entity controller, which implements full CRUD functionality for your entities. Optionally, entities may be created based on classes derived from the provided Entity class.

The entity API cares about creating fieldable entities as well as exportable entities, however it does not yet support revisions. Additionally it supports implementing bundle entities, i.e. bundle objects (like node types) for fieldable entities implemented as (exportable) entities, for which the appropriate field API callbacks get automatically invoked.

For entity types implemented based upon the provided CRUD API the API is providing additional module integration too, i.e. Rules events are provided for all CRUD-related hooks, some basic entity property information for hook_entity_property_info() is provided and exportable entities are automatically integrated with the Features module.

These module integrations are implemented in separate controller classes, which may be separately overridden or enabled/deactivated.
Optionally, the entity API also helps providing an administrative interface for managing entities, e.g. the UI for managing profile types of Profile 2 is built with that.

Rules

The rules module allows site administrators to define conditionally executed actions based on occurring events (known as reactive or ECA rules). It's a replacement with more features for the trigger module in core and the successor of the Drupal 5 workflow-ng module.It allows site administrators to define conditionally executed actions based on occurring events.

Features

Obviously, you may use reaction rules to react upon any event with custom

conditions and actions.
Allows functionality to be re-used via components (Drupal 6: Rule sets only).
Flexible scheduling system that allows scheduling any component / action.
Users can share their customizations by using the built-in import/export feature. For that the module also integrates with Features.
Modular input evaluation system - for example you can install the Token module and use it in every action.
The module has been developed with site performance in mind, so it makes use of caching routines to speed up rule evaluation.
Rules 2.x (Drupal 7 only) features improved APIs, a new admin UI, support for all entity types, parameter configuration via simple data selection (i.e. just pass node:author as argument) and much more. See this blog post for more details.

Taxonomy Manager

This module provides a powerful interface for managing taxonomies. A vocabulary gets displayed in a dynamic tree view, where parent terms can be expanded to list their nested child terms or can be collapsed.

The Taxonomy Manager has following operations and key features:

- dynamic treeview
- mass deleting
- mass adding of new terms
- moving of terms in hierarchies
- merging of terms (using the Term merge module in 7.x)
- fast weight changing with up and down arrows (and AJAX saving)
- AJAX powered term editing form
- simple search interface
- CSV Export of terms
- i18n support for multilingual vocabularies (per language terms)
- Double Tree interface for moving terms in hierarchies, adding new translations and switching terms between different vocabularies

For using the Taxonomy Manager you should have JavaScript and automatically load of images enabled in your browser.

This is a Google Summer of Code 2007 project. Read my proposal and my status reports for more information.

Taxonomy Manager Drupal 5 and JQuery Upadate module:
In case you are using the JQuery Update module v2.x on your D5 installation, the Taxonomy Manager won't work. See this issue for more information.
To fix the problem download the tree.js file and replace it with the one included in the taxonomy manager folder under /js.

Taxonomy Manager Drupal 6 and Panels 3 / Taxonomy Breadcrumb:

Taxonomy Manager and other taxonomy modules might conflict, if both try to override the taxonomy/term/% path. Taxonomy Manager uses this to redirect previously merged terms to their new term. If you do not need this feature or if you want to used Panels 3 / Taxonomy Breadcrumb, you can disable it under Administer > Site Configuration > Taxonomy Manager. See this issue for more information.
Note: since 6.x-2.1, the redirect of taxonomy/term is disabled by default

Taxonomy subterms

Taxonomy Subterms is a simple module to force display child taxonomy terms on the parent taxonomy page of nodes belonging to the current term or/and the children of the current term.

Taxonomy subterms makes use of the exact same page callback of the default taxonomy terms page (or the one you put in place).
The module just add the depth param with an 'all' value to the original arguments and pass them to the "normal" page callback.

Ubercart

Ubercart out of the box is a great solution for stores that sell physical goods, digital product downloads, and site memberships.

Current Features:

Configurable product catalog includes catalog pages and a block to display product categories.
Flexible product creation system with product classes.
Multiple product image support out of the box.
Flexible product attributes system.
Basic product stock level tracking and notification.
Product features to add file downloads, role promotions, and more to products.
Single page checkout.
Automatic account generation (anonymous checkout).
Customer and administrator checkout notifications.
Simple order processing (with workflow for automated order processing).
Simple order creation and editing.
Integrated payment system that acts as a bridge between acceptable payment methods (check, credit card, purchase order, etc.) and payment gateways (CyberSource, Authorize.net, PayPal, etc.).
Shipping quotes and fulfillment, including integration with UPS, FedEx, USPS.
Sales, product, and customer reports.
Activity logging.
Much more... and more to come.

Ubercart is PayPal certified and integrates PayPal Website Payments Standard, Website Payments Pro, and Express Checkout. For more information, you can read the Ubercart PayPal information page.

Views

This module provides a flexible method for Drupal site designers to control how lists and tables of content are presented.

When installing Views on Drupal 7, you should have the current stable version of CTools (currently rc1) and you may need to flush cache as much as twice. There are currently issues with core's cache flushing that means one cache flush isn't necessarily enough.
What is Views

The Views module provides a flexible method for Drupal site designers to control how lists and tables of content (nodes in Views 1, almost anything in Views 2) are presented. Traditionally, Drupal has hard-coded most of this, particularly in how taxonomy and tracker lists are formatted.

This tool is essentially a smart query builder that, given enough information, can build the proper query, execute it, and display the results. It has four modes, plus a special mode, and provides an impressive amount of functionality from these modes.

Among other things, Views can be used to generate reports, create summaries, and display collections of images and other content.
You need Views if

You like the default front page view, but you find you want to sort it differently.
You like the default taxonomy/term view, but you find you want to sort it differently; for example, alphabetically.
You use /tracker, but you want to restrict it to posts of a certain type.
You like the idea of the 'article' module, but it doesn't display articles the way you like.
You want a way to display a block with the 5 most recent posts of some particular type.
You want to provide 'unread forum posts'.
You want a monthly archive similar to the typical Movable Type/Wordpress archives that displays a link to the in the form of "Month, YYYY (X)" where X is the number of posts that month, and displays them in a block. The links lead to a simple list of posts for that month.

Views can do a lot more than that, but those are some of the obvious uses of Views.
Dependencies

The Drupal 7 version of Views requires the Chaos Tool Suite also known as CTools.

Views' bug squad

The Views' bug squad is a group of people who have dedicated a few hours a week to help provide support and fix bugs in the issue queue. They are the first line for answering questions and properly filing issues. Anyone who wishes to join the Bug Squad to help others may do so to help contribute to the Drupal community!
Views documentation

Views 1
Views 2 (This link goes to the Advanced Help project page. All Views documentation is included with its installation. For your convenience, you may also visit http://views-help.doc.logrus.com. This site may or may not be fully up to date at any given time.)
Views 2 API (doxygen generated from Views 2 source. Only useful to developers.)
Views 3 documentation hasn't been written yet. Items of note are:
Only UID1 can import a View.

EXTERNALMODULE

shopmenu

Provideds shopping cart menu and contains ubercart module altered forms.

the "menu" hook (shop_menu). For each merchant from whom I'm using their datafeed, I make code like this in the shop_menu function: function shop_menu($may_cache) { $items = array();   if ($may_cache) { ....