



# DRUPALCON LONDON

# Entities - Emerging Patterns of Usage

---

Presented by Ronald Ashri (ronald\_istos)



@ronald\_istos



I S T O S

Web App Development  
Travel / Semantic Web  
[www.istos.it](http://www.istos.it)



Solutions for hotels,  
villas, B&Bs  
[drupalrooms.com](http://drupalrooms.com)  
w/ Bluespark Labs



Drupal Trainer network  
[drupalteachers.com](http://drupalteachers.com)  
w/ Brightlemon



# Building with Entities

---

- ❖ How we got to entities
- ❖ What do they look like, what can we do, what's missing
- ❖ When and how should we use them
  - ❖ Learning from how they are used so far
- ❖ Questions / Next Steps







slice up a drupal  
and you will find nodes





drupalistas love making lots of nodes and  
mixing them up in different ways  
*... much like pasta*



# Everything is a node



Submitted by **robertDouglass** on Mon, 2006-05-01 06:59

[metadata](#) · [node](#)



- Users, comments, taxonomy terms and attachments become nodes
- Why do this:
  - Simplicity, consistency and code reuse
  - Allow us to extend these content types as we do for nodes, without rewriting code
  - Allow relationships between these, without needing a mapping table
  - Why users:
    - Use CCK and taxonomy instead of profile.module

they love them so much they nearly turned  
everything into a node



user profiles = [drupal.org/project/content\\_profile](https://drupal.org/project/content_profile)



taxonomy = drupal.org/project/taxonomy\_node



comments = drupal.org/project/comment\_nodes





nodes come in different shapes but still tied to  
the same methods and db structure  
*=> need for a more generic core-based solution*





but the core development process  
is never simple



Huge driver turned out to be the  
Field API work -  
attaching fields to core "stuff" came first  
and the need to streamline followed



*“loadable thingies,  
that can be **optionally** fieldable”  
(<http://drupal.org/node/460320>)*



*entités*





Entities in Drupal 7 are the nodes we *really* wanted but didn't know it yet

The *stuff / content* that makes up our site  
*nodes, users, taxonomy terms, comments, files*

User

Node

Taxonomy Term

Comment

Taxonomy  
Vocabulary

File





## Drupal Core Developers Summit 2040

Entities bring us closer to Drupal the framework  
No mention in UI - almost no mention in changelog



# CHANGELOG.TXT...

“In addition, any other object type may register with Field API and allow custom data fields to be attached to itself.”

*object type = entity*



The *relationships* between Entities - the ways entities can interact with each other - define our Drupal application

***nodes can have comments***

**users author nodes**

**taxonomy terms are attached to nodes**

*these relationships are implicit -  
perhaps they could be made explicit?*





how to cook - (or build entities)



# Your Table

## PostIt

postit_id	name	property1	property2	property3

+ fields

(drupal worries about fields)



*you define an entity via a hook\_entity\_info (big array!):*

```
function postit_entity_info(){
  $postit_info['postit'] = array(
    'label' => t('PostIt Note'),
    'controller class' => 'PostItController',
    'base table' => 'postit',
    'uri callback' => 'postit_uri',
    'fieldable' => TRUE,
    'entity keys' => array(
      'id' => 'pid',
    ),
    'static cache' => TRUE,
    'bundles' => array(
      'postit'=> array(
        'label' => 'PostIt',
        'admin' => array(
          'path' => 'admin/structure/postit/manage',
          'access arguments' => array('administer postits'),
        ),
      ),
    ),
    'view modes' => array(
      'full' => array(
        'label' => t('Full PostIt'),
        'custom settings' => FALSE,
      ),
    ),
  );
  return $postit_info;
}
```



```
function postit_entity_info(){  
  $postit_info['postit'] = array(  
    'label' => t('PostIt Note'),  
    'controller class' => 'PostItController',  
  );  
}
```

You can provide you own controller class - typically extending the default DrupalDefaultEntityController which worries about caching, querying, revisioning and attaching fields to entities

```
  ),  
  ),  
),  
'view modes' => array(  
  'full' => array(  
    'label' => t('Full PostIt'),  
    'custom settings' => FALSE,  
  ),  
)  
);  
return $postit_info;  
}
```



```
function postit_entity_info(){
    $postit_info['postit'] = array(
        'label' => t('PostIt Note'),
        'controller class' => 'PostItController',
        'base table' => 'postit',
        'uri callback' => 'postit_uri',
        'fieldable' => TRUE,
        'entity keys' => array(
```

base table = where our main entity data lives  
uri callback = where to look up - view entities  
fieldable = are fields to be attached to our entity

```
    ),
    ),
    'view modes' => array(
        'full' => array(
            'label' => t('Full PostIt'),
            'custom settings' => FALSE,
        ),
    )
);
return $postit_info;
}
```



```
function postit_entity_info(){
  $postit_info['postit'] = array(
    'label' => t('PostIt Note'),
    'controller class' => 'PostItController',
    'base table' => 'postit',
    'uri callback' => 'postit_uri',
    'fieldable' => TRUE,
    'entity keys' => array(
      'id' => 'pid',
    ),
    'static cache' => TRUE,
    'bundles' => array(
      'postit'=> array(
        'label' => 'PostIt',
        'admin' => array(
          'path' => 'admin/structure/postit/manage',
          'access arguments' => array('administer postits'),
        ),
      ),
    ),
  ),
}
```

Bundle = Entity + Configuration + Fields  
1 entity (Node) can have many bundles (Article, Page)



## tell drupal how to load your entity

```
function postit_load($pid = NULL, $reset = FALSE){  
  $pids = (isset ($pid) ? array($pid) : array());  
  $postit = postit_load_multiple($pids, $reset);  
  return $postit ? reset ($postit) : FALSE;  
}
```

```
function postit_load_multiple($pids = array(), $conditions = array(), $reset = FALSE){  
  return entity_load('postit', $pids, $conditions, $reset);  
}
```

```
function entity_load($entity_type, $ids = FALSE, $conditions = array(), $reset = FALSE) {  
  if ($reset) {  
    entity_get_controller($entity_type)->resetCache();  
  }  
  return entity_get_controller($entity_type)->load($ids, $conditions);  
}
```



# ready to cook!

---

- ❖ we also get a query API **EntityFieldQuery**
  - ❖ conditions on properties and fields
  - ❖ queries across entity types
- ❖ and we get **hooks**
  - ❖ `hook_entity_load()`
  - ❖ `hook_entity_presave()`
  - ❖ `hook_entity_insert()`
  - ❖ `hook_entity_update()`
  - ❖ `hook_entity_delete()`
  - ❖ `hook_entity_prepare_view`
  - ❖ `hook_entity_view()`





*Young module developer* : Where is full CRUD, UI,  
Views integration, Tokens, Search, etc?



**Drupal Core:** What? Am I supposed to  
do everything around here?



# + Entity Module

([drupal.org/project/entity](http://drupal.org/project/entity))

---

- ❖ Supports full CRUD
- ❖ Quickly provide new entity types
- ❖ Standardised way of working with entity data
- ❖ Integration into drupal ecosystem (Tokens, Views, Rules, Exportables, Search API, RestWS, VBO)
- ❖ Full Tests





so what are we cooking - and how?  
and why?



entities are a uniquely *drupal* concept

**your stuff**

(your db table or something like that)

+

**Field API**

+

**hooks in the drupal world**



if your module / app is introducing data  
you should really be asking yourself  
***why not via entities***

you **get plugged in to the drupal world**  
for (almost) ***zero cost*** and  
**your data can remains as is (almost)**





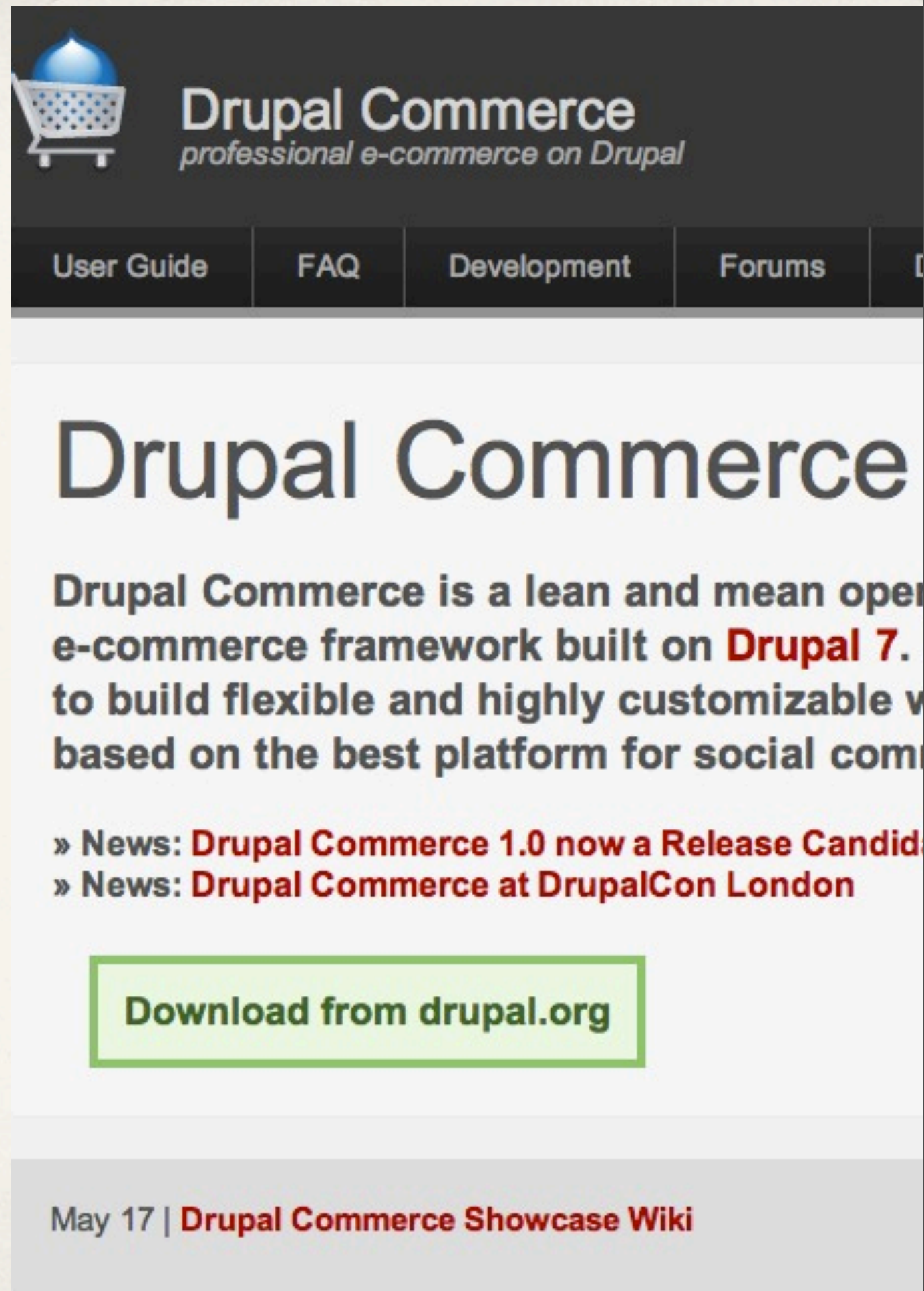
emerging patterns



# Commerce

---

- ❖ Great example as there is also historical perspective
- ❖ D6 Ubercart did almost everything “outside” of Drupal
- ❖ D7 Commerce is a fully signed up member to the Drupal ecosystem



The screenshot shows the Drupal Commerce website. At the top, there is a dark header with a shopping cart icon containing a blue flame, the text "Drupal Commerce" in white, and the tagline "professional e-commerce on Drupal" in a smaller font. Below the header is a navigation bar with links: "User Guide", "FAQ", "Development", "Forums", and a partially visible "D". The main content area has a large heading "Drupal Commerce" followed by a paragraph: "Drupal Commerce is a lean and mean open e-commerce framework built on **Drupal 7**. to build flexible and highly customizable v based on the best platform for social com". Below this are two news items: "» News: **Drupal Commerce 1.0 now a Release Candidate**" and "» News: **Drupal Commerce at DrupalCon London**". A green button with the text "Download from drupal.org" is centered below the news. At the bottom, a grey footer bar contains the text "May 17 | **Drupal Commerce Showcase Wiki**".



in order to add commerce capabilities we need to  
introduce a whole set of concepts (and their supporting  
data structures)

Drupal Commerce defines

Customer Profiles

Lines Items

Orders

Payment Transactions

Products

as entities



Drupal Commerce defines *only* as much as it needs in terms of fields attached to these entities -

Application core concepts (e.g. sku for product) live in entity base table not via field

More complex core data (e.g. price) gets added via a *locked field*


Allows the user add extra fields where it makes sense




local action


entity types

Home » Administration » Store

Products 

LIST **PRODUCT TYPES**

 Product saved.

 Add a product

SKU	TITLE	TYPE	PRICE	OPERATIONS
SKU1234	Product 1	Product	\$1,000.00	<a href="#">edit</a> <a href="#">delete</a>
SKU12341	Product 12	Product	\$1,000.00	<a href="#">edit</a> <a href="#">delete</a>

Entity  
Listing as a View

operational links



Home » Administration » Store » Orders

Order 1

VIEW

EDIT

PAYMENT

LINE ITEMS

REMOVE	TITLE	SKU	UNIT PRICE	QUANTITY	TOTAL
<input type="checkbox"/>	Product 1	SKU1234	1000.00 USD	1	\$1,000.00

Product

Add line item

BILLING INFORMATION

Country

Italy

Full name

Ronald

Street

Via Street

Residence, building, ...

Floor, stairway, ...

Suburb, post box, ...

Postal Code

97222

City

Modica

☐ Delete this profile

Order status

Pending

User information

Owned by Anonymous

Order history

Created 08/20/2011 - 18:31

Status

Pending

Update log message

Order Entity

Contains Line Item Entities

Related to payment entities

addressfield field

looks familiar!



*use entities to introduce new data types that can be user-extensible*

*provide minimum conf required - easily customisable in install profiles or via custom modules*

*ps: did I mention - you really, really, really need a good reason not to use entities when introducing new data (and data tables) to Drupal*



# Profile 2

---

- ❖ Separates users from descriptions of users
- ❖ Uses different bundles to describe different aspects of the same user
- ❖ Use entities to provide profile level type permissions

## Profile 2

[View](#)[Version control](#)

Posted by [joachim](#) on *November 4, 2009 at 2:16pm*

Designed to be the successor of the core profile module, which in contrast to the deprecated module this module provides a new, flexible way of using the power of fields!

### Requirements

- [Entity API](#) – Version 1.0 **beta 9** or later!

### Comparison to user account fields

As described in the [documentation](#) Drupal 7 allows adding fields to user accounts, which serves as a simple way for building user profiles. The main differences are:

- With profile2 user account settings and user profiles are combined. With the "Profile pages" module enabled users get two separate "My profile" pages.
- Profile2 allows for creating multiple profile types, which may have different permissions (e.g. a general profile + a customer profile)
- Profile2 supports private profile fields, which are only shown to the user and to administrators.

### Relationship to content profile

With the new concept of entities in Drupal 7, this module supersedes the content profile module. However for any situations where profiles as nodes might be needed for Drupal 7 will build upon profile2.

### Features

- The module is aimed to provide a simple, but extensible solution



placed under  
admin/structure

Entity Module UI  
for handling types

Home » Administration » Structure

## Profile types

[+ Add profile type](#) [+ Import profile type](#)

LABEL	STATUS	OPERATIONS					
Main profile (Machine name: main)	Custom	<a href="#">edit</a>	<a href="#">manage fields</a>	<a href="#">manage display</a>	<a href="#">clone</a>	<a href="#">delete</a>	<a href="#">export</a>

Home » Administration » Structure » Profile types

## Edit Main profile

[EDIT](#) [MANAGE FIELDS](#) [MANAGE DISPLAY](#)

**Label \***  
 Machine name: main [\[Edit\]](#)  
The human-readable name of this profile type.

☒ Show during user account registration.

☒ Provide a separate page for editing profiles.  
If enabled, a separate menu item for editing the profile is generated and the profile is hidden from the user account page.

[Save profile type](#) [Delete profile type](#)

per type  
configuration



*use new entity relationships to extend the  
application*

*provide configuration via entity types*



# Organic Groups

---

- ❖ Organic Groups uses Entities because via the Entity API it gets CRUD, Views, Rules integration etc for free
- ❖ Groups are still attached to nodes
- ❖ Automatically created via an indirect user action

## Organic groups

[View](#)[Version control](#)[Revisions](#)

Posted by [Amitaibu](#) on *November 24, 2004 at 11:22pm*

7.x-1.1 Release candidate

Note: OG7 is very different from OG6. See [this](#) video learn more.

Overview

Enable users to create and manage their own 'group' maintains a group home page where subscribers can

They do so by posting the usual node types: blog, site home page which facilitates these posts. The block a group.

Groups may be selective or not. Selective groups require or even invitation -only groups. There are lots of pre

Groups get their own theme, language, taxonomy, and Views module

Example sites

1. <https://community.openatrium.com> – intranet



# Group Entity

Home » Add content

## Create Group

**Title \***

**Group type \***

☐ Not a group type

☒ Group type

**Body (Edit summary)**

when you create a node as a Group Type this triggers the creation of the Group Entity

*separation of concerns opens up interesting possibilities and enables things that were not possible before like better internationalization support*



# Group Membership Entity

## Let us find examples

View

Edit

Outline

Devel

Submitted by [ronald](#) on Sun, 08/21/2011 - 08:07

Examples help us learn

**Groups audience:**  
[Entity Appreciation Club](#)

content

group

	<b>id</b> The group membership's unique ID.	<b>name</b> Reference to a group membership type.	<b>etid</b> The entity ID.	<b>entity_type</b> The entity type (e.g. node, comment, etc').	<b>gid</b> The group's unique ID.	<b>state</b> The state of the group content.	<b>created</b> The Unix timestamp when the group content was created.
<input type="checkbox"/>  	3	og_membership_type_default	4	node	2	1	1313906862



because relationships are entities we can  
add fields to them

*e.g. date field indicating relationship  
expiration*



*use Entities to separate concerns*

-

*using the Field API as a way to flexibly add  
access to configuration options*

-

*Site builder can decide how much config to make  
available to Site Admins*



# Rooms

- ❖ Hotel booking application
- ❖ Introduces Bookable Units and Bookings as entities
- ❖ Relates entities to non *entity-able* data
- ❖ Uses entity types for default values settings as well as a source of common data across all entities instances

## Rooms

[View](#)[Version control](#)[Edit](#)[Revisions](#)[Ma](#)

Posted by [ronald\\_istos](#) on *June 1, 2011 at 3:02pm*

Drupal Rooms is a booking and room management for hotels, vacation rentals and B&Bs.

It allows an accommodation owner to:

- Create rooms and assign availability (or on hold)
- Define seasons and assigned prices for rooms
- Create bookings and keep customer records.
- Make rooms available for booking with a available handling both immediate bookings and booking

Drupal Rooms is being developed as Drupal 7 only

Visit [Drupal Rooms](#) for more.

Project commissioned and sponsored by [Bluespark](#)

## News

**Working towards an alpha?**

With development going a want to follow along you



Project Information

**drupalrooms**  
for hotels, villas and b&bs

Maintenance status: [Active](#)



# Bookable Unit Entity

[Home](#) » [Administration](#) » [Rooms](#) » [Rooms](#)



EDIT

MANAGE AVAILABILITY

MANAGE PRICING

## Bookable Unit Name \*

## Unit Base Price

Base price is per unit per night

☒ Bookable

To remove unit from bookable units uncheck – this will not alter availability states but remove the unit from any searches.

## Sleeps

## Bookable Unit Default State



Choose what state to put the unit in for the dates chosen above

## Discount for single occupancy



# Bookable Unit Availability

[Home](#) » [Administration](#) » [Rooms](#) » [Bookable Units](#) »



EDIT

MANAGE AVAILABILITY

MANAGE PRICING

## Level 1 Rooms - 1 Availability View

[Back](#)

Current

[Forward](#)

### August 2011

SUN	MON	TUE	WED	THU	FRI	SAT
31	1	2	3	4	5	6
	AV					
7	8	9	10	11	12	13
N/A		AV				
14	15	16	17	18	19	20
AV						
21	22	23	24	25	26	27
All Sorts						AV
28	29	30	31	1	2	3
AV						
4	5	6	7	8	9	10

### September 2011

SUN	MON	TUE	WED	THU	FRI	SAT
28	29	30	31	1	2	3
				AV		
4	5	6	7	8	9	10
AV						
11	12	13	14	15	16	17
AV						
18	19	20	21	22	23	24
AV						
25	26	27	28	29	30	1
AV						
2	3	4	5	6	7	8

### October 2011

SUN	MON	TUE	WED	THU	FRI	SAT
25	26	27	28	29	30	1
						AV
2	3	4	5	6	7	8
AV						
9	10	11	12	13	14	15
AV						
16	17	18	19	20	21	22
AV						
23	24	25	26	27	28	29
AV						
30	31	1	2	3	4	5
AV						

## Update Room Availability

Start Date \*

Format: 2011/08/21

End Date \*

Format: 2011/08/21

Bookable Unit State


Choose what state to put the unit in for the dates chosen above

Update Room Availability



# Entity Types define default property values

[Home](#) » [Administration](#) » [Rooms](#) » [Bookable Unit Types](#)

Edit Standard Doubles 

EDIT

MANAGE FIELDS

## Label \*

Machine name: standard\_doubles [[Edit](#)]

The human-readable name of this unit type.

## Unit Base Price

Base price is per unit per night – the price you set here will be the default price for all units of this type. You can then base price on a per unit basis

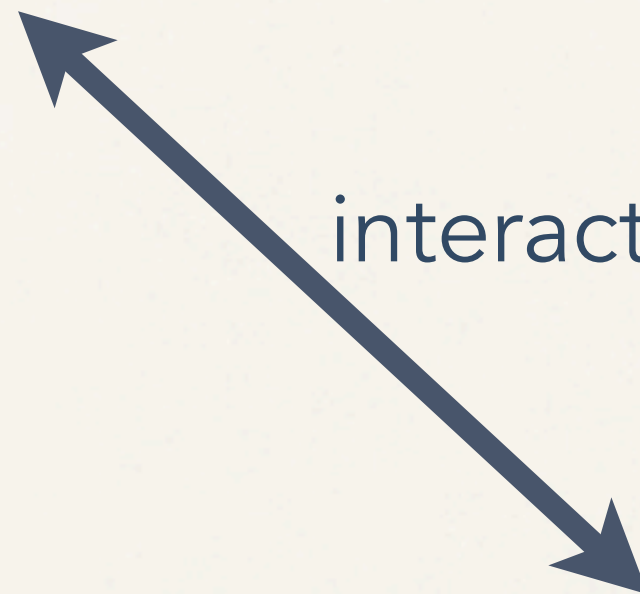
## Sleeps

The default number of people that can sleeps in units of this type

Save unit type



Rooms



interact via entities

Commerce



# Kickstarting Development

entities in a jar





# Model Entities

Implementation of a generic Entity and Entity Administration interface that you can use to kickstart your own project.

[drupal.org/project/model](http://drupal.org/project/model)

## Model Entities

[View](#) [Version control](#) [Edit](#) [Revisions](#) [Maintainers](#)

Posted by [ronald\\_istos](#) on May 8, 2011 at 10:24am

The objective of this module is to provide a model entity *and* entity administration interface to kick-start your entity development.



A screenshot of the 'Model Entities' module's entity list interface. It shows a table with columns for 'Entity ID', 'Name', and 'Description'. The table contains three rows of data.

Entity ID	Name	Description
1	Entity 1	
2	Entity 2	
3	Entity 3	



A screenshot of the 'Model Entities' module's structure interface. It shows a list of modules with their descriptions and a 'Your Drupal' link.

Module	Description
Model Entities	Configure and manage model entities in your Drupal site.
Entity Admin	Manage entity administration, including entity types, fields, and relationships.
Entity Types	Manage entity types, including adding and removing fields and the display of fields.
Entity Fields	Manage entity fields, including adding and removing fields and the display of fields.
Entity Relationships	Manage entity relationships, including adding and removing relationships and the display of relationships.
Entity Views	Manage entity views, including adding and removing views and the display of views.

While entities in Drupal 7 are a very useful abstraction and can be great tool for suitable projects there are some challenges facing developers wanting to use them:

- Identifying what is a suitable project for entities.
- Figuring out what is "best practice" when it comes to entity development and **quickly getting up to speed with working code.**
- Effectively integrating entities with Drupal core and modules that are entity-aware and can give you extra functionality for free.

These challenges are partially being met by:



# experiment and learn from others

---

doing it your way is great...

but also need to study and share patterns  
for doing things





# Summary

---

- ❖ Drupalize your data with entities
- ❖ Improves distinction between drupal the framework and drupal the CMS improving app development
- ❖ Leverage the entity module
- ❖ **Learn from examples**





@ronald\_istos



I S T O S

Web App Development  
Travel / Semantic Web  
[www.istos.it](http://www.istos.it)



Solutions for hotels,  
villas, B&Bs  
[drupalrooms.com](http://drupalrooms.com)  
w/ Bluespark Labs



Drupal Trainer network  
[drupalteachers.com](http://drupalteachers.com)  
w/ Brightlemon



# What did you think?

Locate this session on the  
DrupalCon London website:

<http://london2011.drupal.org/conference/schedule>

Click the “Take the survey” link

***THANK YOU!***

