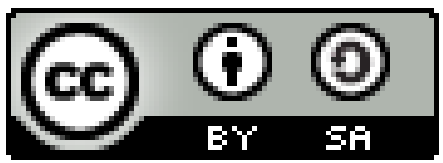
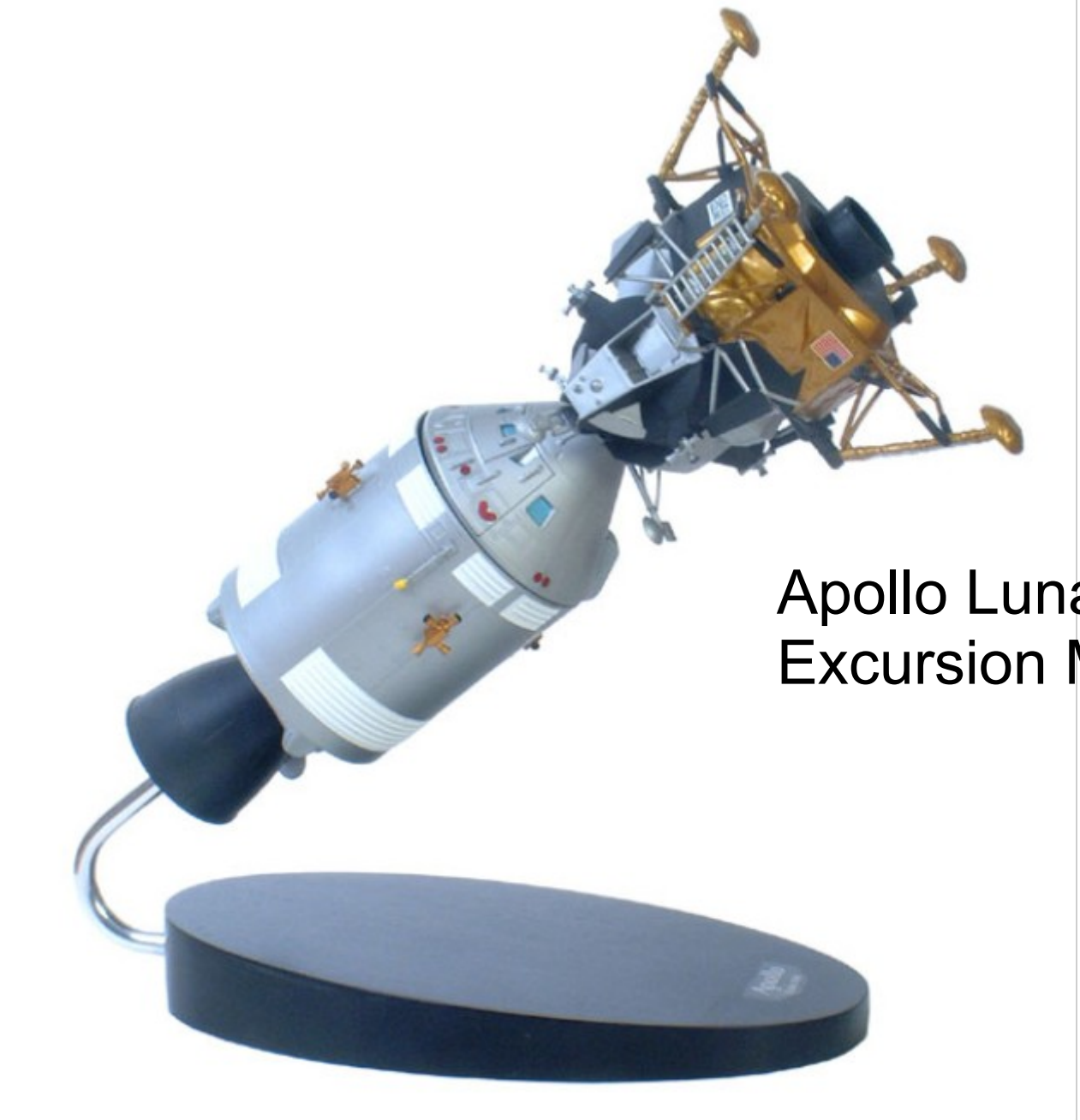
Introduction to Module Development

Ezra Barnett Gildesgame
Growing Venture Solutions
@ezrabg on Twitter
ezra-g on Drupal.org

DrupalCon Chicago 2011



What is a module?



Apollo Lunar Service and Excursion Modules

What is a module?

A piece of software that adds or extends functionality on a Drupal website

Adds functionality

- Ubercart - Provides a sophisticated online store
- Fivestar - Provides a star rating widget for rating content
- Signup - Let's users sign up to attend events

Extends Functionality

Integration:

- **UC_Signup** - Allows people to to pay for Ubercart Events

Utility:

- **Token** - Many modules rely on Token to provide configurable strings (like "Hello, [user-name]")

Plugin:

- **GMap Views** – Display information on a Google Map.

Often, adding a new feature entails extending or connecting existing features.

You usually don't have to start from scratch :)

- Fivestar **stores data using the VotingAPI**
- UC_Signup **connects Ubercart and Signup**
- Embedded Media Field **provides a new field for use with the Content Construction Kit**

Core, Contrib & Custom

- Core - Part of the official Drupal package. Everyone using Drupal has these
- Contrib - Publicly available optional download, not specific to a particular website. (Though sometimes specific to a kind of feature)
- Custom - Specific to a particular website. Useful for that website, but not generally useful to the public.

Custom Modules

- Often avoidable (Extend existing contrib instead)
- Sometimes necessary, good for site-specific tweaks

- Usually cost more to maintain. You alone deal with:
  - Drupal API changes
  - Security issues
  - Feature additions
  - Friends you lose when you duplicate their module :(
  - Feeling of utter embarrassment when you spend hours writing a module only to discover that many people use a well-regarded contrib alternative. You have wasted your time only to create something vastly inferior to a project that is the result of thousands of hours of worldwide community collaboration. You spend weeks sulking in social isolation eating only nachos and briefly consider switching to Ruby but take time off from development entirely to focus on feng shui.

You usually don't have to start from scratch!

Drupal Core and Contrib have powerful systems that you can harness in your module:

- Displaying and processing forms for user input
- Handling user accounts
- Handling Content (Core node, contrib CCK)
- Creating custom listings of content (Views)

And much, much more!

http://www.flickr.com/photos/redherring1up/163079104/

How do modules add or extend functionality?

Hooks

I'm performing a particular action.
Does anyone have anything to say about it?

Person A: I'm getting up to get napkins. Does anybody want me to get anything else?

Person B: Yes! Please get straws.

Module A: I'm presenting a form to the user.

Module B: Please add a checkbox to the form!

A - The hook Definition
B - The hook Implementation

Hook Events and Example Actions

- **Displaying a form to the user (hook_form_alter)**
  - Add a custom checkbox
- **A user signs up for an account (hook_user)**
  - Display a friendly message
- **Submitting a node (hook_nodeapi)**
  - Store custom data in the database
- **The website sends mail (hook_mail)**
  - Connect to an SMTP sever

Common module pattern

- Implement one or more hooks
- Use a Drupal API helper function
- Add custom processing

Form API

PHP


```php
$form['your-name'] = array(
  '#type' => 'textfield',
  '#title' => t('Your Name'),
  '#default_value' => account['name'],
  '#size' => 60,
  '#maxlength' => 64,
  '#description' => t('Enter your real name here.'),
);
```

HTML

```html
<div class="form-item" id="edit-your-name-wrapper">
 <label for="edit-your-name">Your Name: </label>
 <input type="text" maxlength="64" name="your-name" id="edit-your-name"
size="60" value="" class="form-text" />
 <div class="description">Enter your real name here.</div>
</div>
```

Form API

- Forms described consistently as PHP arrays
- Control over access, validation & submission
- Easy to modify forms
- Added security

How to get started?

- Read about one or more hooks
- Find an example module
- Dive in!

Hi, Drupal! I'm a module:

- .info file – Tells Drupal that your module exists.

- .module file – Contains main module code

Optional

- .install file
- .inc files

Hooks Demo

- Client: American Society for Delicious Foods

Hooks Demo

- Client: American Society for Delicious Foods (ASDF)
- ASDF Module

Desired feature:

Redirect user to appropriate category (taxonomy term) page after submitting a recipe.

Tools we'll use:

- hook_form_alter()
- Devel module's dpm() function

Documentation!
Drupal API Reference & Form API Quickstart Guide

dpm(): Our flashlight into Drupal

Easily print any variable to the screen for inspection.

dpm($input, $name = NULL)

I'm using mah dpm() flashlight to inspect ur Drupal variabulz. http://www.flickr.com/photos/moogan/55851439/

Dive in!

- Goggles optional

Our approach:

- Target a specific from
- Add a custom submit handler to the “Save” button
- In custom submit handler, set $form_state['redirect'] based on submitted taxonomy term value
- Rejoice

variable_get($name, $default)

Retrieve a site-wide variable.

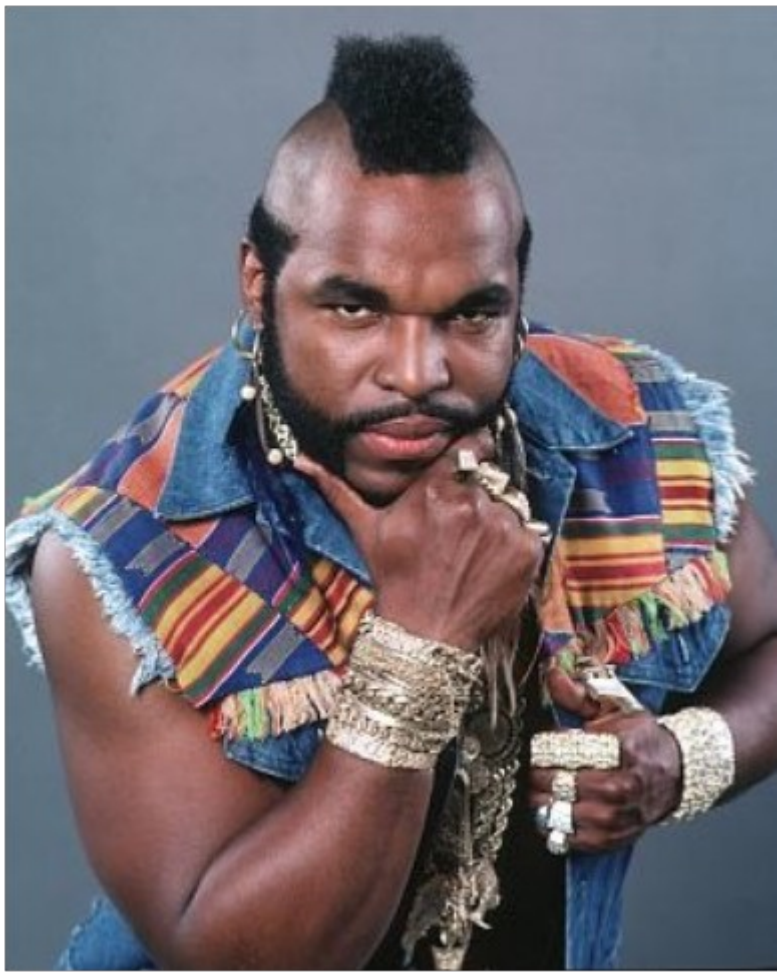
also, variable_set()

user_access($string, $account = NULL)

Check if a user has permission to perform a particular action.

t()

Translate Interface Text.



“I translate the fool!”

hook_menu()

Map a function to a callback:

http://example.com/your-callback
Makes Drupal run a particular function.

Common module pattern

- Implement one or more hooks
- Use a Drupal API helper function
- Add custom processing

Look inside your favorite module!

You *can* handle it!

```
75  /**
76   * Insert our checkbox, add a submit button, and populate fields.
77   */
78  function comment_notify_form_alter(&$form, &$form_state, $form_id) {
79    global $user;
80
81    // Only alter the form if it's a comment form and the user has the permission to subscribe.
82    if ($form_id == 'comment_form' && (user_access('subscribe to comments') || user_access('administer comments'))) {
83      // Only add the checkbox if this is an enabled content type
84      $node = node_load($form['nid']['#value'] ? $form['nid']['#value'] : $form['nid']['#default_value']);
85      $enabled_types = variable_get('comment_notify_node_types', drupal_map_assoc(array($node->type)));
86      if (empty($enabled_types[$node->type])) {
87        return;
88      }
89
90      $available_options = _comment_notify_options();
91
92      drupal_add_css(drupal_get_path('module', 'comment_notify') .'/comment_notify.css');
93      drupal_add_js(drupal_get_path('module', 'comment_notify') .'/comment_notify.js');
```

Debugging/Learning tools:

func_get_args()

debug_backtrace()

func_get_args()

func_get_args()

```
404 function user_save($account, $edit = array(), $category = 'account') {
405   $args = func_get_args();
406   dpm($args, 'Args passed to user_save()');
```

See what arguments are passed to a function.
Compare args passed on a successful operation vs an unsuccessful one.

Got WTF? Use debug_backtrace().

```
394 /**
395  * Generates a 403 error if the request is not allowed.
396  */
397 function drupal_access_denied() {
398   // Why are we getting an access denied message?!
399   $backtrace = debug_backtrace();
400   dpm($backtrace, 'Backtrace from drupal_access_denied');
401
402   drupal_set_header('HTTP/1.1 403 Forbidden');
403
```

Resources

- api.drupal.org
- Drupal Form API Reference
- Drupal Form API Quickstart guide
- Pro Drupal Development Book
- Example modules
- GVS API Cheat sheet
- #Drupal on IRC
- Patch contributor guide
- *Look inside your favorite module!*
- @ezrabg on twitter

What did you think?

Submit a session evaluation:

https://chicago2011.drupal.org/node/add/eval/834

Thanks!

DRUPALCON
CHICAGO
MARCH 7-10 2011