

# Drupal 模块开发-API

## -Drupal培训Lesson 4

**Rapid Technologies**

[www.chinaldap.com](http://www.chinaldap.com)

2011-5-14

Rapid Technologies



# 课程安排

- 使用模块来处理node- add, delete,update,view
- 菜单系统 - Menu
- 表单系统 Form API
- 常用Hooks

本课程适合熟悉PHP语言语言而且熟悉模块开发基本概念用户

# 使用模块来处理node



# 问答模块需求与分析

## ✓ 需求

- 用户能发布一个问题，这个问题除了基本的标题，提问的内容，问题创建时间，问题创建者，还需要有个**过期时间**
- 其他用户可以来回答问题，添加自己的答案，答案的属性包括回答者，回答时间，回答的内容。
- 提问者可以从这些答案中，选择一个自己满意的答案，标注为“最佳答案”

## ✓ 分析

- 有个**node type : question** ,用户来发布问题
- 有个**node type : answer**, 用户来提交答案
- 需要一个数据库表，把问题与答案关联起来
- 每个答案上，需要一个**button**,让提问者点击作为一个“最佳答案”
- 。 。 。 其他还有很多细节，暂时不考虑

# 看看drupal的node 属性

Drupal 的默认的一个node ,自动存储了很多属性。  
比如：

- `node->title` - 一个node的标题
- `node->body` - 一个node 的实际内容
- `node->uid` - 一个node 的创建用户的id
- `node->name` - node 的创建用户的名字
- `node->date` - node 的创建日期

想一想我们前一页分析的要创建的两个node type,还缺少什么?

参考node的属性 <http://drupal.org/node/49768>

# 使用模块处理node - 定义数据表

## ✓ question\_answer.install 定义自己的数据表

```
$schema['question'] = array(
    'description' => 'record the extra question fields',
    'fields' => array(
        'q_nid' => array(
            'type' => 'int',
            'unsigned' => TRUE,
            'not null' => TRUE,
            'default' => 0,
            'description' => 'the question {node}.nid',
        ),
        'select_a_nid' => array(
            'type' => 'int',
            'unsigned' => TRUE,
            'not null' => TRUE,
            'default' => 0,
            'description' => 'the selected answer {node}.nid',
        ),
        'duration' => array(
            'type' => 'int',
            'unsigned' => TRUE,
            'not null' => TRUE,
            'default' => 0,
            'description' => 'The question expire duration',
        ),
    ),
    'primary key' => array('q_nid')
);
```

```
$schema['answers'] = array(
    'description' => 'record the answers for questions',
    'fields' => array(
        'sid' => array(
            'type' => 'serial',
            'size' => 'medium',
            'unsigned' => TRUE,
            'not null' => TRUE,
            'description' => t("The serial number for a question answer section."),
        ),
        'q_nid' => array(
            'type' => 'int',
            'unsigned' => TRUE,
            'not null' => TRUE,
            'default' => 0,
            'description' => 'the question {node}.nid',
        ),
        'a_nid' => array(
            'type' => 'int',
            'unsigned' => TRUE,
            'not null' => TRUE,
            'default' => 0,
            'description' => 'the answer {node}.nid',
        ),
    ),
    'primary key' => array('sid')
);
```

# 使用模块处理node – 定义node type

## ■ Hook\_node\_info -- 定义node type

```
function question_answer_node_info() {  
    return array( 'questions' => array(  
        'name' => t('Questions'),  
        'module' => 'question_answer',  
        'description' => t('a chemical questions '),  
        'has_title' => TRUE,  
        'title_label' => t('Title'),  
        'has_body' => TRUE,  
        'body_label' => t('Question'),  
    ),  
  
}
```

# 使用模块处理node – 定义 form

- **Hook\_form(\$node)** - 用户提交一个问题的时候，需要一个form

```
function question_answer_form(&$node) {  
    $type = node_get_types('type', $node);  
    if ($type->has_title) {  
        $form['title'] = array(  
            '#type' => 'textfield',  
            '#title' => check_plain($type->title_label),  
            '#required' => true,  
            '#size' => 10,  
            '#default_value' => $node->title,  
            '#weight' => -5,  
        );  
        $form['duration'] = array(  
            '#type' => 'textfield',  
            '#title' => t('Duration'),  
            '#required' => TRUE,  
            '#default_value' => 0,  
            '#weight' => 3  
        );  
    }  
    return $form;}
```

# 使用模块处理node – 存储用户提交的数据

- Hook\_insert() - 存储额外的数据到我们的表里

```
/** * implements hook_insert(). */
function question_answer_insert($node)
{
    db_query('INSERT INTO {question} (q_nid, select_a_nid, duration) ' .
        "VALUES (%d, %d, %d)",
    $node->nid, 0, $node->duration);
}
```

通过这个hook,我们把用户提交的数据存储起来，一部分，提交的title, body等，Drupal自己处理，另外一部分，duration等，就存储到我们自己的表里

# 使用模块处理node – 加载我们的数据

- Hook\_load ,加载自己数据库里的数据。

```
/** * Implementation of hook_load(). *
function question_answer_load($node) {
    $result = db_query('SELECT * FROM {question} WHERE q_nid = %d', $node->nid);
    return db_fetch_object($result);
}
```



往往在调用一个node的数据的时候，使用node\_load(\$nid), 这个时候就触发了我们的这个question\_answer\_load, 所以就把默认的node的属性（title,body等），再加上我们自己的表里的数据 duration等，给合并起来。

# 使用模块处理node – 显示我们的的数据

this is devin first questionn, how old are you



Submitted by [devin](#) on Mon, 02/21/2011 - 11:15

I have to answer: \*

Submit

Other answers

sfasf afasfd

[Use this answer](#)

Mon, 04/04/2011 - 22:47 by devin

what? ?

[Use this answer](#)

Wed, 04/20/2011 - 22:30 by devin

问题  
显示

用户  
提交  
答案

答  
案  
列  
表

# 使用模块处理node – 显示我们的的数据

- **Hook\_view** – 当想显示一个**question**的时候，你可以任意修改这个显示的样子。

```
/** * Implementation of hook_view(). */
function question_answer_view($node, $teaser = FALSE, $page = FALSE) {
  $node = node_prepare($node, $teaser); // get it ready for display

  $node->content['submit_answer'] = array(
    '#value' => drupal_get_form('question_answer_edit_answer_form',
$node->nid),      '#weight' => 10,
);

  $node->content['answers'] = array(
    '#value' => question_answer_answers_list($node->nid),
    '#weight' => 11,
  );
  return $node;
}
```

# 使用模块处理node – 其他。。

- **Hook\_init**, 在每个页面被请求时调用，常用来加载css, javascript
- **Hook\_menu**, 定义菜单, url [\(稍后详解\)](#)
- **Hook\_perm** , 实现自己的权限控制
- **Hook\_access**, 定义了自己模块对node 的处理，包括对update, add, delete, view的操作限制
- **hook\_validate**, 在form提交数据后，可以在这里验证数据的有效性
- **Hook\_delete**, 在删除一个node时候，可以通过这个hook来处理自己的数据，比如删除自己数据表里的数据
- **Hook\_block**, 定义自己的block [\(稍后详解\)](#)
- **Hook\_theme**, 实现自己的theme机制 [\(稍后详解\)](#)

# 菜单系统



# 菜单系统 – 概括

- 主要实现了网站的导航(navigation)系统和回调(call back)
- `Hook_menu` 定义了菜单项目并赋予给了一定的路径，  
drupal根据路径，生成了一个树状的菜单系统。
- 比如,路径 a, a/b, e, a/b/c/d, f/g, and a/b/h

- a
  - a/b
    - a/b/c/d
    - a/b/h
- e
- f/g

# 菜单系统 – hook\_menu

```
/** * Implementation of hook_menu(). */
function question_answer_menu() {
    $items['qa'] = array( //http://hgc27044.chinaw3.com/index.php?q=qa
        'title' => t('Chemical Answers'),
        'page callback' => 'question_answer_list',
        'page arguments' => NULL,
        'access arguments' => array('access content'),
    );
}

$items['qa/resolved'] = array(
    'title' => 'Resolved Questions',
    'page callback' => 'question_answer_list',
    'page arguments' => array('1'),
    'type' => MENU_LOCAL_TASK,
);
```



如果在hook\_menu里没有出现type,则默认是 MENU\_NORMAL\_ITEM

# 菜单系统 – 菜单类型 type

- **MENU\_NORMAL\_ITEM**: 一般的菜单项，可以在整个的菜单树里，通过后台的控制，可以隐藏，移动。
- **MENU\_CALLBACK**: 仅仅注册一个路径，不会出现在菜单树里，后台不能控制，通过url可以访问到.
- **MENU\_SUGGESTED\_ITEM**: 定义了一个“建议”的菜单项，后台可以来enable.
- **MENU\_LOCAL\_TASK**: 定义了一个tab.
- **MENU\_DEFAULT\_LOCAL\_TASK**: 每个local task 集合里，都必须有个默认的，这个和父菜单调用同一个函数.

# 菜单系统 – 一个演示

Tab

The screenshot shows a user interface for managing questions. At the top, there are three tabs: "All Questions" (selected), "Resolved Questions", and "Unresolved Questions". Below the tabs, a question is listed:

如何实现一个自己定义的block ? Unresolved

在drupal 里, 如何实现一个自定义block

Fri, 05/06/2011 - 13:38 by admin

# 需要熟悉的参考资料

- Working with the Drupal API
- <http://drupal.org/node/206753>
- 本课程使用的问答模块代码：  
[http://www.chinaldap.com/sites/chinaldap.com/files/question\\_answer.zip](http://www.chinaldap.com/sites/chinaldap.com/files/question_answer.zip)

# 问题与解答



请把问题或者 [意见发送到 qq1392137041@gmail.com](mailto:qq1392137041@gmail.com), 或者论坛 [www.chinaldap.com](http://www.chinaldap.com)  
或者直接了当加我qq: 1392137041, 或者直接进入 qqtalk房间号码: 6289868