

# 又一年，Uliweb发展纪事

李迎辉(limodou@gmail.com)

2012/10/20

Uliweb 转到github,同时在 [code.google.com](http://code.google.com)

Ver 0.1.6

Uliweb-doc为专门的Uliweb文档项目

Plugs集成越来越多的组件, 包括JQuery, Bootstrap, AngularJS等

Ver 0.1.4

在多个场合介绍Uliweb, 如: 《我们的开源项目》

UliwebZone项目, 由何家胜部署在 <http://uliweb.clkg.org>

2012 [cn.pycon.org](http://cn.pycon.org) 官网开发

# UliwebZone的实战

Uliweb

Plugs

SQLAlchemy

par - Markdown解释器，加入bootstrap的语法支持

JQuery + 插件

AngularJS

# 论坛

Uliweb Zone [首页](#) [论坛](#) [教程](#) [课程](#) [管理](#) [关于](#)  [limodou](#) | [帐户设置](#) | 

2 条新消息 · [点击查看](#) ×

Uliweb	主题	文章	最新文章
<a href="#">Uliweb开发</a> Uliweb开发相关讨论	30	78	<a href="#">limodou &gt;</a> 2周, 5天之前
Plugs	主题	文章	最新文章

# 论坛

The screenshot displays the Uliweb Zone forum interface. At the top, there is a navigation bar with 'Uliweb Zone' and links for '首页', '论坛', '教程', '课程', '管理', and '关于'. A user profile for 'limodou' is visible in the top right corner, along with a notification for '2条新消息, 点击查看 x'. Below the navigation bar, the forum post is titled 'sdfsdfsdf' and has '26篇评论' (26 comments). The first two posts are by 'limodou' and contain the text 'asfsdfsfs'. The first post is dated '2012-03-26 09:59:37 GMT +8' and the second is dated '2012-03-26 10:00:31 GMT +8'. The first reply is also by 'limodou' and is dated '2012-03-26 10:02:04 GMT +8', containing the text 'asdfsfsd'. The second reply is also by 'limodou' and is dated '2012-07-28 13:19:49 GMT +8', containing the text 'asfadsfsfadsf'. A vertical sidebar on the left contains a '分享到...' (Share to...) button.

# 教程

Uliweb Zone 首页 论坛 教程 课程 管理 关于

limodou | 帐户设置 | 退出

2 条新消息, 点击查看 ×

+ 添加新教程

---



## 《Markdown语法说明》

由 limodou 更新, 最后更新时间 0 分钟 之前  
介绍uliwebzone实现的Markdown语法

[点击阅读 >](#)

访问次数: < 29 评论: < 20

---



## 《开源与沉淀》

由 limodou 更新, 最后更新时间 2 月, 4 周 之前  
介绍我的开源历程与我创建的开源项目。

[点击阅读 >](#)

访问次数: < 2 评论: < 0

---

# 教程

The screenshot shows the '教程' (Tutorials) management page on Uliweb Zone. The page has a navigation bar with 'Uliweb Zone' and links for '首页', '论坛', '教程', '课程', '管理', and '关于'. The user 'limodou' is logged in, with a profile picture and a '帐户设置' (Account Settings) link. A notification bubble in the top right corner says '2 条新消息, 点击查看 x'. The main content area has a breadcrumb trail: '返回 / Markdown语法说明 / 目录'. Below this are three action buttons: '+ 增加新的章节' (Add new chapter), '修改教程信息' (Edit tutorial info), and '删除教程' (Delete tutorial). The tutorial list contains three items:

- 1 **Markdown语法说明** [字数:9723/评论数:18/访问次数:21]
- 2 **Test Code** [字数:2763/评论数:2/访问次数:10]
- 3 **新章节** [字数:2014/评论数:0/访问次数:8]



# 教程

The screenshot displays the Uliweb Zone website interface. At the top, there is a navigation bar with the site name 'Uliweb Zone' and menu items: '首页' (Home), '论坛' (Forum), '教程' (Tutorial), '课程' (Course), '管理' (Management), and '关于' (About). A user profile for 'limodou' is visible in the top right corner, with links for '帐户设置' (Account Settings) and a refresh icon.

The main content area shows a breadcrumb trail: '返回 / Markdown语法说明 / Markdown语法说明'. Below this, there are buttons for '2 Test Code', '编辑' (Edit), and '删除' (Delete). The title of the page is 'Markdown语法说明', with a subtitle indicating it was updated by 'limodou' on '2012-09-25 12:03:10+00:00' and has '22' views.

The content begins with a '标题' (Title) section, followed by a 'Setext方式' (Setext style) section. A code block shows the following content:

```
1. 标题1
2. -----
3.
4. 标题2
5. -----
```

On the right side, there is a dark sidebar containing a table of contents with the following items: '标题', 'Setext方式', '标题2', 'Atx方式', '标题2', '段落', '代码段', '引用', '列表', and '分隔线'.

# 课程

Uliweb Zone [首页](#) [论坛](#) [教程](#) **课程** [管理](#) [关于](#) limodou | [帐户设置](#)

2 条新消息, [点击查看](#) x

### 课程列表 [更多»](#)

- Python Web开发**  
简介: Python Web [查看详情»](#)
- Uliweb Blog开发**  
简介: Test [查看详情»](#)
- Uliweb 入门**  
简介: Test [查看详情»](#)
- Linux内核**  
简介: Test [查看详情»](#)

### 最新动态 [更多»](#)

### 讲师介绍



# 课程

Uliweb Zone 首页 论坛 教程 课程 管理 关于 limodou | 帐户设置

2 条新消息, 点击查看 ×

Uliweb Zone 首页 论坛 教程 课程 管理 关于 limodou | 帐户设置

2 条新消息, 点击查看 ×

📁 课件: [查看课件](#)

## 课程安排

第 9 期 • 未开始 已报名

**20** **17**  
招收 剩余

授课讲师: limodou  
上课时间: 2012年10月18日 12点 - 2012年10月27日 12点  
上课地点: bj  
课程性质: 免费  
[报名查看](#)

[取消报名](#)

第 8 期 • 结束

**7** **7**  
招收 剩余

授课讲师: limodou  
上课时间: 2012年9月30日 08点 - 2012年9月30日 08点  
上课地点: dddd [查看地图](#)  
课程性质: 收费 • 50元/人  
[报名查看](#)

# 消息

Uliweb Zone 首页 论坛 教程 课程 管理 关于 limodou | 帐户设置 |

设置  
管理设置  
消息  
**接收消息**  
发送消息  
已发消息  
论坛  
课程

全部 已读 未读 全部类型 全部设为已读 全部删除

- 由 limodou 发送于 2 周, 1 天之前 ✕  
tttt
- 由 limodou 发送于 2 周, 1 天之前 ✕  
test
- 由 test 发送于 2 周, 2 天之前 ✕  
test 评论了教程 《Test Code》  
Test @guest
- 由 test 发送于 2 月, 1 周之前 ✕  
用户 test 在论坛提到了你, [查看帖子](#)
- 由 test 发送于 2 月, 1 周之前 ✕  
test 评论了教程 《Markdown语法说明》

# 管理

Uliweb Zone 首页 论坛 教程 课程 管理 关于 smodou | 帐户设置 |

2 条新消息 · 点击查看 ×

· 设置  
- 管理设置  
  用户管理  
  角色管理  
  权限管理  
- 消息  
  接收消息  
  发送消息  
  已发消息  
  论坛  
- 课程  
  **课程分类**  
  教师管理  
  课程管理

新课程分类名称  +添加

名称	操作
Core	☑ ×
Uliweb	☑ ×
Test	☑ ×
Hello	☑ ×

localhost:8000 上的网页显示：  
请输入新的课程分类名称  
  
确定 取消

localhost:8000/class/admin/category#

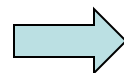
# AngularJS



Javascript MVC框架

基于DOM的模板引擎

双向数据绑定

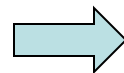


Dirty data check

模块化

控制器

Directive



HTML标签扩展

◦ ◦ ◦

```
{{use "angularjs"}}  
{{use "jquery"}}
```

装入相关的资源

```
<div ng-app="AdminApp" id="ng-app" ng-controller="adminCtrl">  
  <form class="form-inline">  
    <input ng-model="topicitype_name"  
      placeholder="新课程分类名称"> </input>  
    <a class="btn btn-primary" ng-click="add_category()">添加</a>  
  </form>  
  <table class="table table-bordered table-header">  
    <tbody>  
      <tr ng-repeat="c in objects">  
        <td>{% c.name %}</td>  
        <td>  
          <a ng-click="edit_category(c.id)">  
            <i class="icon-edit" title="编辑"></i>  
          </a>  
          <a ng-click="remove_category(c.id)">  
            <i class="icon-remove" title="删除"></i>  
          </a>  
        </td>  
      </tr>  
    </tbody>  
  </table>  
</div>
```

Template



```
{{use "angularjs"}}  
{{use "jquery"}}
```

```
<div ng-app="AdminApp" id="ng-app" ng-controller="adminCtrl">  
  <form class="form-inline">  
    <input ng-model="topictype_name"  
      placeholder="新课程分类名称"> </input>  
    <a class="btn btn-primary" ng-click="add_category()">添加</a>  
  </form>  
  <table class="table table-bordered table-header">  
    <tbody>  
      <tr ng-repeat="c in objects">  
        <td>{% c.name %}</td>  
        <td>  
          <a ng-click="edit_category(c.id)">  
            <i class="icon-edit" title="编辑"></i>  
          </a>  
          <a ng-click="remove_category(c.id)">  
            <i class="icon-remove" title="删除"></i>  
          </a>  
        </td>  
      </tr>  
    </tbody>  
  </table>  
</div>
```

设置App和Controller

Template

```
{{use "angularjs"}}
{{use "jquery"}}

<div ng-app="AdminApp" id="ng-app" ng-controller="adminCtrl">
  <form class="form-inline">
    <input ng-model="topic_type_name"
      placeholder="新课程分类名称"> </input>
    <a class="btn btn-primary" ng-click="add_category()">添加</a>
  </form>
  <table class="table table-bordered table-header">
    <tbody>
      <tr ng-repeat="c in objects|orderBy:Ordering">
        <td>{% c.name %}</td>
        <td>
          <a ng-click="edit_category(c.id)">
            <i class="icon-edit" title="编辑"></i>
          </a>
          <a ng-click="remove_category(c.id)">
            <i class="icon-remove" title="删除"></i>
          </a>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

绑定变量

Template

```
{{use "angularjs"}}
{{use "jquery"}}

<div ng-app="AdminApp" id="ng-app" ng-controller="adminCtrl">
  <form class="form-inline">
    <input ng-model="topictype_name"
      placeholder="新课程分类名称"> </input>
    <a class="btn btn-primary" ng-click="add_category()" >添加 </a>
  </form>
  <table class="table table-bordered table-header">
    <tbody>
      <tr ng-repeat="c in objects|orderBy:Ordering">
        <td>{% c.name %}</td>
        <td>
          <a ng-click="edit_category(c.id)">
            <i class="icon-edit" title="编辑"> </i>
          </a>
          <a ng-click="remove_category(c.id)">
            <i class="icon-remove" title="删除"> </i>
          </a>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```



绑定click事件



Template

```
{{use "angularjs"}}  
{{use "jquery"}}
```

```
<div ng-app="AdminApp" id="ng-app" ng-controller="adminCtrl">  
  <form class="form-inline">  
    <input ng-model="topicname" placeholder="新课程分类名称"> </input>  
    <a class="btn btn-primary" ng-click="add_category()">添加</a>  
  </form>  
  <table class="table table-bordered table-striped">  
    <tbody>  
      <tr ng-repeat="c in objects|orderBy:Ordering">  
        <td>{% c.name %}</td>  
        <td>  
          <a ng-click="edit_category(c.id)">  
            <i class="icon-edit" title="编辑"></i>  
          </a>  
          <a ng-click="remove_category(c.id)">  
            <i class="icon-remove" title="删除"></i>  
          </a>  
        </td>  
      </tr>  
    </tbody>  
  </table>  
</div>
```

对objects进行循环

在DOM显示变量值(表达式)

绑定click事件进行编辑

绑定click事件进行删除

Template

```
var app = angular.module('AdminApp', []).config(function ($interpolateProvider) {
    $interpolateProvider.startSymbol('{%');
    $interpolateProvider.endSymbol('%}');
});
function adminCtrl($scope) {
    $.get('/class/admin/category/query')
        .success(function(data, status, headers, config){
            $scope.objects = data.rows;
            $scope.$apply();
        });
    $scope.ordering = 'name';
    $scope.add_category = function(){
        var v = $('#field_category').val();
        if (!v){
            show_message('课程分类名称不能为空', 'error');
        }else{
            $.post('/class/admin/category/add', {'name':v})
                .success(function(data){
                    if (data.success) $scope.objects.push(data.data);
                    else show_message(data.message, 'error');
                    $scope.$apply();
                });
        }
    }
}
```

初始化angularjs标签

Template

```
var app = angular.module('AdminApp', []).config(function ($interpolateProvider) {
  $interpolateProvider.startSymbol('{%');
  $interpolateProvider.endSymbol('%}');
});
function adminCtrl($scope) {
  $.get('/class/admin/category/query')
  .success(function(data, status, headers, config){
    $scope.objects = data.rows;
    $scope.$apply();
  });
  $scope.ordering = 'name';
  $scope.add_category = function(){
    var v = $('#field_category').val();
    if (!v){
      show_message('课程分类名称不能为空', 'error');
    }else{
      $.post('/class/admin/category/add', {'name':v})
      .success(function(data){
        if (data.success) $scope.objects.push(data.data);
        else show_message(data.message, 'error');
        $scope.$apply();
      });
    }
  }
}
```

从后台装入数据(只会执行一次)

Template

```
var app = angular.module('AdminApp', []).config(function ($interpolateProvider) {
  $interpolateProvider.startSymbol('{%');
  $interpolateProvider.endSymbol('%}');
});
function adminCtrl($scope) {
  $.get('/class/admin/category/query')
    .success(function(data, status, headers, config){
      $scope.objects = data.rows;
      $scope.$apply();
    });
  $scope.ordering = 'name';
  $scope.add_category = function(){
    var v = $('#field_category').val();
    if (!v){
      show_message('课程分类名称不能为空', 'error');
    }else{
      $.post('/class/admin/category/add', {'name':v})
        .success(function(data){
          if (data.success) $scope.objects.push(data.data);
          else show_message(data.message, 'error');
          $scope.$apply();
        });
    }
  }
}
```

设置排序字段

Template

```
var app = angular.module('AdminApp', []).config(function ($interpolateProvider) {
  $interpolateProvider.startSymbol('{%');
  $interpolateProvider.endSymbol('%}');
});
function adminCtrl($scope) {
  $.get('/class/admin/category/query')
    .success(function(data, status, headers, config){
      $scope.objects = data.rows;
      $scope.$apply();
    });
  $scope.ordering = 'name';
  $scope.add_category = function(){
    var v = $('#field_category').val();
    if (!v){
      show_message('课程分类名称不能为空', 'error');
    }else{
      $.post('/class/admin/category/add', {'name':v})
        .success(function(data){
          if (data.success) $scope.objects.push(data.data);
          else show_message(data.message, 'error');
          $scope.$apply();
        });
    }
  }
}
```

Template

添加数据处理，界面自动刷新



```
#coding=utf8  
from uliweb import expose, functions  
from uliweb.orm import get_model
```

导入常用函数

```
@expose('/class/admin/category')  
class ClassCategoryAdminView(object):  
    def __init__(self):  
        self.model = get_model('class_category')  
  
    def __begin__(self):  
        functions.require_login()  
  
        if not functions.has_role(request.user, 'superuser'):  
            error("你没有权限访问此页面")  
  
    def query(self):  
        view = functions.ListView(self.model, pagination=False)  
        return json(view.json())  
  
    def add(self):  
        view = functions.AddView(self.model, success_data=True)  
        return view.run(json_result=True)
```

View

```
#coding=utf8  
from uliweb import expose, functions  
from uliweb.orm import get_model
```

```
@expose('/class/admin/category')
```

```
class ClassCategoryAdminView(object):  
    def __init__(self):  
        self.model = get_model('class_category')
```

```
    def __begin__(self):  
        functions.require_login()
```

```
        if not functions.has_role(request.user, 'superuser'):  
            error("你没有权限访问此页面")
```

```
    def query(self):  
        view = functions.ListView(self.model, pagination=False)  
        return json(view.json())
```

```
    def add(self):  
        view = functions.AddView(self.model, success_data=True)  
        return view.run(json_result=True)
```

设置Class的URL

View

```
#coding=utf8
from uliweb import expose, functions
from uliweb.orm import get_model


@expose('/class/admin/category')
class ClassCategoryAdminView(object):
    def __init__(self):
        self.model = get_model('class_category')

    def __begin__(self):
        functions.require_login()

        if not functions.has_role(request.user, 'superuser'):
            error("你没有权限访问此页面")

    def query(self):
        view = functions.ListView(self.model, pagination=False)
        return json(view.json())

    def add(self):
        view = functions.AddView(self.model, success_data=True)
        return view.run(json_result=True)
```



初始化



View

```
#coding=utf8
from uliweb import expose, functions
from uliweb.orm import get_model

@expose('/class/admin/category')
class ClassCategoryAdminView(object):
    def __init__(self):
        self.model = get_model('class_category')

    def __begin__(self):
        functions.require_login()

    if not functions.has_role(request.user, 'superuser'):
        error("你没有权限访问此页面")

    def query(self):
        view = functions.ListView(self.model, pagination=False)
        return json(view.json())

    def add(self):
        view = functions.AddView(self.model, success_data=True)
        return view.run(json_result=True)
```

执行Class方法 前的处理

View

```
#coding=utf8
from uliweb import expose, functions
from uliweb.orm import get_model

@expose('/class/admin/category')
class ClassCategoryAdminView(object):
    def __init__(self):
        self.model = get_model('class_category')

    def __begin__(self):
        functions.require_login()

        if not functions.has_role(request.user, 'superuser'):
            error("你没有权限访问此页面")

    def query(self):
        view = functions.ListView(self.model, pagination=False)
        return json(view.json())

    def add(self):
        view = functions.AddView(self.model, success_data=True)
        return view.run(json_result=True)
```

View

返回数据处理

```
#coding=utf8
from uliweb import expose, functions
from uliweb.orm import get_model

@expose('/class/admin/category')
class ClassCategoryAdminView(object):
    def __init__(self):
        self.model = get_model('class_category')

    def __begin__(self):
        functions.require_login()

        if not functions.has_role(request.user, 'superuser'):
            error("你没有权限访问此页面")

    def query(self):
        view = functions.ListView(self.model, pagination=False)
        return json(view.json())

    def add(self):
        view = functions.AddView(self.model, success_data=True)
        return view.run(json_result=True)
```

View

新增的处理

## 代码框架自动生成

```
D:\project\mywork\uliwebzone>uliweb generic
Appname: classes
Table Name: class_info
View Class Name [Class_infoView]:
Save views to [views_class_info.py]:
Class View URL prefix [/classes]: /classes/info
Creation Theme([a]ngularjs): a
```

处理常见的： List, Add, Edit, Delete, View

目前支持angularjs，html等以后还要支持其它模式

谢谢！明年再见！