

**江苏物联网研究发展中心**

**数据云服务平台后台管理系统**

**代码说明**

**所属项目编号：**

**所属项目名称：**

**所属项目负责人：**

**报告撰写人：王煜**

**报告版本号：V1.0**

**数据交易与服务技术研究中心**

**2013年 05月03 日**

**表1：版本历史**

|  |  |  |
| --- | --- | --- |
| **版本** | **日期** | **描述** |
| V1.0 | 2013-05-03 | 初始版本 |
|  |  |  |
|  |  |  |

**目 录**

[一、说明 5](#_Toc358472245)

[1.1编写目的 5](#_Toc358472246)

[1.2约定和限制 5](#_Toc358472247)

[二、运行环境 6](#_Toc358472248)

[三、系统架构 6](#_Toc358472249)

[3.1 系统设置部分 7](#_Toc358472250)

[3.2 db模块 9](#_Toc358472251)

[3.3 app模块 9](#_Toc358472252)

[3.4 templates文件夹 10](#_Toc358472253)

[3.5 static文件夹 10](#_Toc358472254)

[3.6 utils文件夹 10](#_Toc358472255)

[四、app部分 10](#_Toc358472256)

[4.1 productManagement 10](#_Toc358472257)

[4.2 purchaseManagement 12](#_Toc358472258)

[4.3 sysAdminManagement 12](#_Toc358472259)

[4.4 tariffManagement 13](#_Toc358472260)

[4.5 transactionBillManagement 13](#_Toc358472261)

[4.6 userManagement 14](#_Toc358472262)

[五、utils文件夹 15](#_Toc358472263)

#### 一、说明

##### 1.1编写目的

本文档是对数据云服务平台后台管理系统的代码进行说明，供有关人员阅读、评定。

##### 1.2约定和限制

该文档只适用于数据服务云平台后台管理系统项目。

#### 二、运行环境

该系统为B/S系统，运行在linux系统上，编程语言为python、html，web架构是django，数据库用mysql。

系统访问地址是<http://IP:8000/>

用户名：密码是root:23456

#### 三、系统架构

该系统采用的是django架构，结构如下：



其中，billing子目录下的settings.py、urls.py和wsgi.py是系统的设置部分。

Static文件夹存放系统的图片、js文件、css文件。

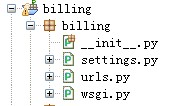
templates文件夹是系统的模板文件。

Db模块是对数据库的一些操作。

其它的是编写的app模块，即平台系统的各功能模块。

Utils定义了一些公用的功能。

##### 3.1 系统设置部分



**1）urls.py**

该系统的url设置,这里该文件只是一个基础的定义，每个app的url定义通过include函数分派到各自app下的urls.py文件中。

urlpatterns = patterns(*''*,

(r*'^$'*,*'login.views.index'*),

(r*'^login/'*,include(*'login.urls'*)),

(r*'^usersManagement/'*,include(*'usersManagement.urls'*)),

(r*'^my/messages/'*,include(*'messagesManagement.urls'*)),

(r*'productManagement/'*,include(*'productManagement.urls'*)),

(r*'purchaseManagement/'*,include(*'purchaseManagement.urls'*)),

(r*'sysAdminManagement'*,include(*'sysAdminManagement.urls'*)),

(r*'tariffManagement/'*,include(*'tariffManagement.urls'*)), (r*'transactionBillManagement/'*,include(*'transactionBillManagement.urls'*)),

)

**2）settings.py**

该系统的设置和配置。

#数据库设置

DATABASES = {

*'default'*: {

*'ENGINE'*:*'django.db.backends.mysql'*,

*'NAME'*:*'zx'*,

*'USER'*: *'root'*,

*'PASSWORD'*: *'root0423'*,

*'HOST'*: *''*,

*'PORT'*: *''*,

}

}

#模板位置

TEMPLATE\_DIRS = (

os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_),os.path.pardir,*'templates'*)),

)

#app设置

INSTALLED\_APPS = (

……

# 自己编写的app

*'login'*,

*'usersManagement'*,

*'messagesManagement'*,

*'productManagement'*,

*'tariffManagement'*,

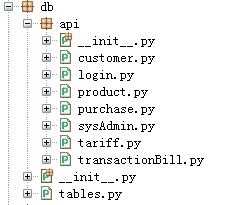
*'purchaseManagement'*,

*'transactionBillManagement'*,

*'sysAdminManagement'*,

)

##### 3.2 db模块



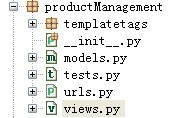
Db模块是对数据库的操作。

tables.py定义了该系统的数据库模型，它相当于django定义中的models.py。这里定义了该系统数据库所有数据结构。

Api子模块里是各app对数据库的所有操作，比如添加、修改、删除信息等。

##### 3.3 app模块

每个app模块都有以下几个文件构成：



templatetags是该模块的模板库。tables.py是数据在数据库中的定义。urls.py 是该访问该模块的url描述。views.py 视图，即业务逻辑描述。其具体内容在第四部分说明。Models.py文件是空的，因为我们已经在db.tables.py里定义模型了。

##### 3.4 templates文件夹



templates是模板文件夹，一个app模板一个子文件夹。

##### 3.5 static文件夹



Static里放了定义模板页面需要的css文件、图片和javascrip文件。

##### 3.6 utils文件夹

codeDocImg06.jpg

Apps都需要的功能在这里定义。

#### 四、app部分

下面将针对每个app做分别说明。

##### 4.1 productManagement

该app是对产品以及相关信息的管理

**1）views.py**

|  |  |
| --- | --- |
| serviceCategorysIndex(request): | 列出所有服务类型（云服务、增值服务等） |
| addServiceCategory(request) | 新建服务类型 |
| editServiceCategoryForm(request) | 服务类型的编辑表单 |
| editServiceCategory(request) | 更新服务类型信息 |
| deleteServiceCategory(request) | 删除服务类型。如果该类型已有下属的产品种类，则不能删。 |
| productCategoryIndex(request) | 列出所有产品种类（云主机、云存储等） |
| addProductCategory(request) | 新建产品种类 |
| editProductCategoryForm(request) | 产品种类的编辑表单 |
| editProductCategory(request) | 更新产品种类信息 |
| deleteProductCategory(request) | 删除产品种类。如果该种类已有产品，不能删。 |
| productIndex(request) | 列出所有产品信息 |
| addProductForm(request) | 新建产品表单 |
| addProduct(request) | 新建产品 |
| productDetail(request) | 产品的详细信息 |
| editProductForm(request) | 产品信息的编辑表单 |
| editProduct(request) | 更新产品信息 |
| inactiveProduct(request) | 注销该产品 |

##### 4.2 purchaseManagement

该app是对产品交易的管理

**1）views.py**

|  |  |
| --- | --- |
| transactionIndex(request) | 列出所有用户购买的所有产品 |
| search(request) | 条件搜索产品 |

##### 4.3 sysAdminManagement

系统管理员信息的管理。

**1）views.py**

|  |  |
| --- | --- |
| adminGroupsIndex(request) | 列出所有管理员级别 |
| addAdminGroup(request) | 新建管理员级别 |
| editAdminGroupForm(request) | 管理员级别信息的编辑表单 |
| editAdminGroup(request) | 更新管理员级别信息 |
| delAdminGroup(request) | 删除某个管理员级别 |
| sysAdminIndex(request) | 列出所有系统管理员 |
| addSysAdmin(request) | 新建系统管理员 |
| editSysAdminForm(request) | 管理员信息的编辑表单 |
| editSysAdmin(request) | 更新管理员信息，其中超级管理员不能删除 |
| delSysAdmin(request) | 删除某系统管理员 |

##### 4.4 tariffManagement

资费管理

**1）views.py**

|  |  |
| --- | --- |
| discountsIndex(request) | 优惠信息列表 |
| addDiscount(request) | 新建优惠 |
| editDiscountForm(request) | 优惠信息编辑表单 |
| editDiscount(request) | 更新优惠信息 |
| inactiveDiscount(request) | 注销该优惠 |
| packagesIndex(request) | 套餐列表 |
| addPackageForm(request) | 新建套餐的表单 |
| addPackage(request) | 新建套餐 |
| editPackageForm(request) | 套餐信息编辑表单 |
| editPackage(request) | 更新表单信息 |
| inactivePackage(request) | 注销该套餐 |

**2）templatetags.tariff\_transform.py**

|  |  |
| --- | --- |
| transform\_productDetail(parser,token) | 将产品id转化成产品名 |

##### 4.5 transactionBillManagement

产品购买账单管理

**1）views.py**

|  |  |
| --- | --- |
| billIndex(request) | 所有用户的账单列表 |
| search(request) | 条件搜索账单 |
| accountBillList(request) | 根据帐户列出相关账单 |

##### 4.6 userManagement

客户管理

**1）views.py**

|  |  |
| --- | --- |
| customerIndex(request) | 客户列表 |
| customerDetail(request) | 客户的详细信息 |
| EditCustomerForm(request) | 编辑客户信息表单 |
| editCustomer(request) | 更新客户信息 |
| delCustomer(request) | 注销客户 |
| accountIndex(request) | 客户的帐号列表 |
| customerGroupsIndex(request) | 客户级别列表 |
| addCustomerGroupForm(request) | 客户级别新建表单 |
| addCustomerGroup(request) | 新建客户级别 |
| editCustomerGroupForm(request) | 客户级别信息编辑表单 |
| editCustomerGroup(request) | 更新客户级别信息 |

**2）templatetags.cusGroup\_transform.py**

|  |  |
| --- | --- |
| trans\_discount(parser,token) | 根据优惠值得id取优惠值 |
| trans\_product(parser,token) | 根据产品id取产品名 |

#### 五、utils文件夹

一些apps使用到的功能函数

**1）common.py**

|  |  |
| --- | --- |
| get\_currentId(url) | 抓取url中的id号 |