

10 数据模型

- [信息类型](#)
- [数据建模](#)

信息类型

Drupal将CMS系统中的所有数据，分为四类（CMS数据）：

- 内容Content：显示在系统中的主要数据，如新闻、产品、文件、图片等；
- 配置Configuration：控制系统行为、名称、提示等，如块怎么放置、站点名等等；
- 会话Session：系统和用户之间的交互数据，如IP地址、浏览器信息、登录时间等；
- 状态State：当前系统的状态，通常是机器自动产生的、人类不可编辑的一些信息，允许丢失，比如计划任务最后允许的时间。

针对上述四类信息，Drupal有不同的存储机制。

用的最多的“内容”和“配置”数据被进一步地抽象和归纳出实体概念。

数据建模

- 概念：即依据要管理的信息，为其设计信息数据的结构模型，此模型涉及到数据如何储存，程序系统对信息的管理建立在此模型基础之上，是系统设计的前期工作。
- 数据建模是一个CMS系统和内容管理系统需要提供的基础功能之一，也是最重要的功能之一。在CMS系统中只需要使用系统提供的管理界面进行配置建表，不需要手动构建。CMS系统会依据使用者的配置自动在数据库中建相应的数据表和字段。
- Drupal对信息的抽象--实体Entity：数字世界中的一个实实在在的、客观存在的、能感受到的主体。
 - 实体概念（用户角度）：
 - 实体可以看做是一个信息主体、一个信息单元，或一条结构化的信息，它包含了组成自身的各种信息。如一个用户是一个实体，它包含了用户名、密码、注册时间、邮件地址、头像等。在Drupal中实体有两大类型--内容实体和配置实体，区别在于：
 - 内容实体通常是系统内需要显示的内容信息，如新闻、咨询、评论、用户等；数据是表结构，由字段组成；
 - 配置实体通常用于存储配置信息，如站点配置、验证码配置等；数据是树型结构，可以有多个层级。
 - 实体概念（开发者角度）：
 - Drupal是面向对象编程的，即OOP开发，实体是一个对象，代表一个结构化数据，该对象中储存了一个信息主体的所有信息，以及和这些信息相关的操作方法：一个表示“新闻”的实体对象，其每一个信息，如：正文、简介、作者、发布时间等都储存在该对象的一个属性中，被

称为字段属性，同时实体对象有save()、create(\$value = [])、delete()等等方法，这些方法可以操作实体中的数据。

- 官方解释：实体是一个用来持久化储存内容或配置的对象。
- 实体字段：
 - 概念：
 - 字段用于储存实体的某一方面的信息
 - 字段具有类型，比如文字、日期、数字、布尔、图片等等
 - 模块可以扩展，以提供更多自定义的字段类型
 - 不同类型的字段有不同的设置，储存不同的信息
 - 实体字段与数据库字段：两者都有类型概念，数据库字段只是数据库表中的一个列，而实体字段概念更加高级（重要）：
 - 实体字段可以有多个属性，每个属性可以对应一个数据库字段，比如常用的图像字段有alt、title等
 - 一个自定义的实体字段，在底层时用一整张数据表来储存
 - 实体字段可以没有数据库字段与之对应，即不储存在数据库中
 - 实体字段可以进行配置，比如是否必填、是否可翻译、默认值、描述等等
 - 实体字段可以是虚拟的，其由程序提供，并没有实际的可储存的数据，比如“伪字段”，在某些高级用法中使用
 - 实体字段可以是实时计算出来的，即可计算字段，它的值是在需要的时候直接计算出来的，并不是来自于储存
 - 实体字段可以是只读的
 - 实体自行负责数据的具体储存、加载，提供接口给其他程序调用
 - 实体及实体字段实际上提供了一个数据层，向上层系统屏蔽了数据是如何储存的，使得系统架构简洁、灵活
 - 实体字段中的信息通常保存在数据库中，但也可以保存在其他储存器中，比如文件中
 - 内容实体和配置实体的区别：
 - 内容实体对象中，每个字段属性都对应一个字段对象，字段对象是实体对象的属性。
 - 在Drupal中的数据建模通常操作的就是内容实体，我们为内容实体创建不同的字段，用来表示我们所需要的信息，创建字段的过程就是在做数据建模，在系统中使用最多的就是“节点实体”（node），为什么叫节点？源于历史上Drupal用节点来代表系统中管理的内容，节点实体对应着后台管理界面的“内容”，我们在发布一篇文章时，就是在建立一个节点实体。
 - 内容实体的数据结构：实体 --> 字段列表1,2,3 --> （每个字段列表对应的）字段1,2,3 --> （每个字段对应的）字段属性1,2,3。
 - 内容实体类型：内容实体可以有多种类型，除了节点实体外，默认提供：内容（node）、用户（user）、自定义区块（block content）、评论（comment）、

联系信息 (contact message)、文件 (file)、媒体 (media)、URL别名 (path alias)、快捷方式链接 (shortcut)、分类术语 (taxonomy term)、定制菜单链接 (menu link content)。

- 在每一种内容实体类型中，每个实体都有一个ID标识，用于区别该类型下的不同实体个体，不同实体类型还有一些通用属性，比如：语言代码、是否可翻译、标题、版本标识等。
- 内容实体子类型Bundle：
 - 每种内容实体可以有多个子类型，即Bundle。比如节点实体默认提供了“文章”和“基本页面”，这些bundle就是节点实体的子类型，即内容类型。
 - 内容实体可以有0或多个Bundle，即可以不要子类型，比如系统中的账户实体就没有子类型。
 - 所有Bundle共享的字段成为基本字段，是这个实体类型的通用字段，通常放在一张数据库表中储存，Bundle可以有自己特有的字段，称为Bundle字段，该种字段在各Bundle间通常是不共享的，但也可以共享。
- 配置实体对象是对PHP多维关联数组的封装，用于储存配置信息，提供对配置的操作方法。
 - 在用户使用层面上，对配置实体的感受没有那么明显，主要是因为配置实体不涉及数据建模，它的结构由底层开发人员定义，用户只是在使用它，一般不会关注其内部实现。特点如下：
 - 配置实体没有字段概念，也没有Bundle概念
 - 并非所有配置数据都以配置实体方式保存，它们也可以以配置对象的方式保存，这称为简单配置
 - 在最底层，所有配置实体是建立在配置对象之上的，配置对象负责持久化保存配置数据，为什么需要进一步包装成配置实体？主要是因为其和内容实体实现相同的接口，从而达到使用上的统一
 - 配置实体储存的配置数据是树型结构，在取、存某个配置值时，是通过“树”中的路径来实现的
 - 配置实体和内容实体一样，起到提供一个数据层的目的，上层系统因此无需知道数据具体是如何储存的，这种分层式设计提供很多灵活性

- Drupal中其他数据模型：

- 在Drupal中，默认以实体方式储存内容数据，数据建模以内容实体为基础。
- 虽然Drupal大量运用实体，但也并非一切都是实体，比如键值储存、翻译数据、日志数据、缓存系统等。
- 模块依然可以直接面向数据库，直接使用自定义数据库表，直接调用数据库操作接口，这得益于系统的分层设计，实体是比直接数据库操作更加高一层的数据封装层，因此Drupal可以容纳任意数据架构模型。