相比于Flask等Python Web框架，Django的功能是最完整的，定义了服务发布，路由映射，模板编程，数据处理的一整套功能，各个模板紧密耦合(对于高级开发人员而言并不好，无法进行轻量级开发，但对于初学者很友好)。

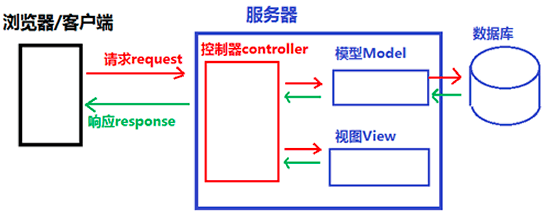
Django的设计模式是MTV，即Model-Template-View，在Web开发中还有另外一种设计模式MVC(Model-View-Controller)。MTV模式是从MVC中衍生出来的。

MVC模式：

Model：数据存储层，定义数据表并进行增删改查

View：视图层，系统前端显示部分，负责显示什么和如何进行显示

Controller：控制层，负责根据从View层输入的指令来检索Model层的数据，编写代码产生结果并输出



具体的请求和响应：

1. 客户端发出请求request到Controller，Controller接受请求后，分别向Model和View发出指令
2. Model层根据指令与数据库交互检索数据，然后将数据返回给Controller
3. View层根据指令检索页面，并将此页面发送给Controller
4. Controller接受了Model和View的结果后，将其组织成响应格式，然后发送到客户端，客户端解析后展示出来。

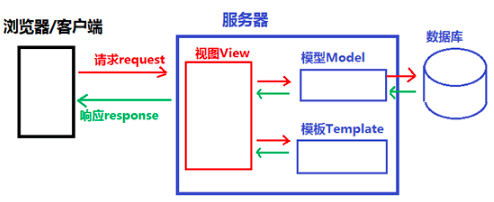
MVC的3层紧密相连，又相互独立，每层的修改都不会影响其他层，每层都提供了独立的接口供其它层调用。降低了耦合性，提高了可重用性。

Django的MTV模式：

Model：数据存储层，类似MVC中的Model层

Template：模板层，也叫表现层，处理页面的显示

View：业务逻辑层，处理具体的业务，联通Model和Template



具体的请求和响应：

1. 客户端发送request请求，View层解析请求后，同时向Model层和Template层发送指令
2. Model层根据指令同数据库交互，并将数据返回给View层
3. Template层接受指令后，调用相应的模板返回给View层
4. View层接受了数据和模板后，首先渲染模板(将相应的数据赋值给模板)，然后组织成响应格式，发送给客户端

对比MVC模式，用View取代了Controller，并用Template取代了原来的View。

Django中的Tempalte可以理解为一个HTML，它的渲染在View层。

MTV是对MVC的细化，将MVC中的V进行拆分，视图的显示和如何显示交给了T，V层更专注业务逻辑。

Django其实是有Controller层的，但是有框架实现的，所以开发者不用关心。

Django Rest-framework中MTV模式，后端工程师其实只实现M和V，T是由前端工程师实现，以保证前后端分离。

Django中的app的概念：在Django中，project指的是整个Web项目/系统，app指的是这个项目中的模块组件，负责实现项目的一部分功能。例如一个电子商务系统，它是一个project，而账户体系、支付系统、商品体系等模块，就是不同的app。

一个Django项目的基本结构

