**一、简述**

在软件开发中，MVC与三层架构这两个专业词汇经常耳闻，同时总有很多人将它们混为一谈，认为三层架构就是指MVC，给它画上等号，但实际上，这是错误的认知，并不是说它们没有任何关系，而是MVC与三层架构不是简单的相等。下面将拿javaweb开发中的MVC（SSM框架）与三层架构进行比较，让大家理清两者之间的关系。

**二、概念**

**1、系统架构**

所谓系统架构是指整个应用系统程序大的结构，常见的系统架构有三层架构与MVC。前面已经说了，三层架构与MVC不是简单的相等，它们存在差别，但又联系。现在可以肯定的是，这两种系统架构的出现，都是为了降低系统模块间的耦合度。

**2、三层架构**

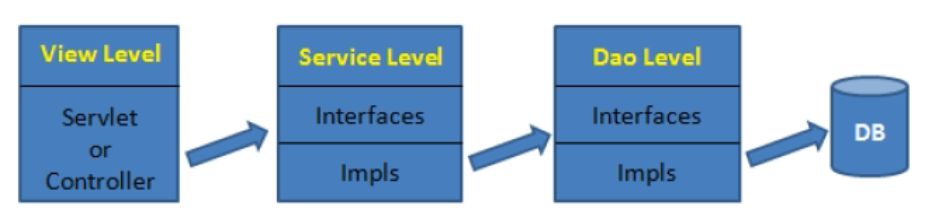
三层架构是指：视图层View、服务层Service、持久层Dao，分别完成不同的功能。

View层：用于接收用户提交请求的代码在这里编写。

Service层：系统的业务逻辑主要在这里编写。

Dao层：直接操作数据库的代码在这里编写。

为了更好的降低各层间的耦合度，在三层架构程序设计中，采用面向抽象编程。即上层对下层的调用，是通过接口实现的。而下层对上层的真正服务提供者，是下层接口的实现类。服务标准（接口）是相同的，服务提供者（实现类）可以更换。这就实现了层间的耦合。



三层架构

**3、MVC**

MVC是指：Model模型、View视图、Controller控件器。

View：视图，为用户提供使用界面，与用户直接进行交互。

Model：模型，承载数据，并对用户提交请求进行计算的模块。其分为两类，一类称为数据承载Bean，一类称为业务处理Bean。所谓数据承载Bean是指实体类，专门承载业务数据的，如Student、User等。而业务处理Bean则是指Service或Dao对象，专门用于处理用户提交请求的。

Controller：控制器，用于将用户请求转发给相应的Model进行处理，并处理Model的计算结果向用户提供相应响应。

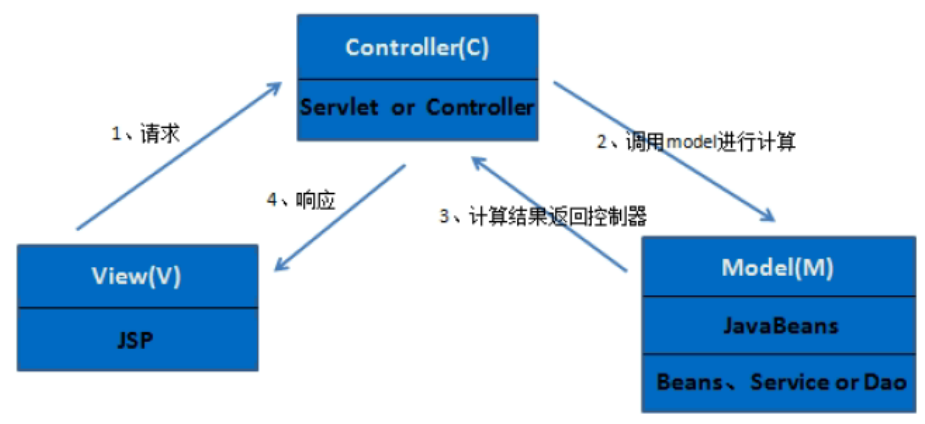
MVC架构程序的工作流程是这样的：

（1）用户通过View页面向服务端提出请求，可以是表单请求、超链接请求、AJAX请求等。

（2）服务端Controller控制器接收到请求后对请求进行解析，找到相应 的Model对用户请求进行处理。

（3）Model处理后，将处理结果再交给Controller。

（4）Controller在接到处理结果后，根据处理结果找到要作为向客户端发回的响应View页面。页面经渲染（数据填充）后，再发送给客户端。



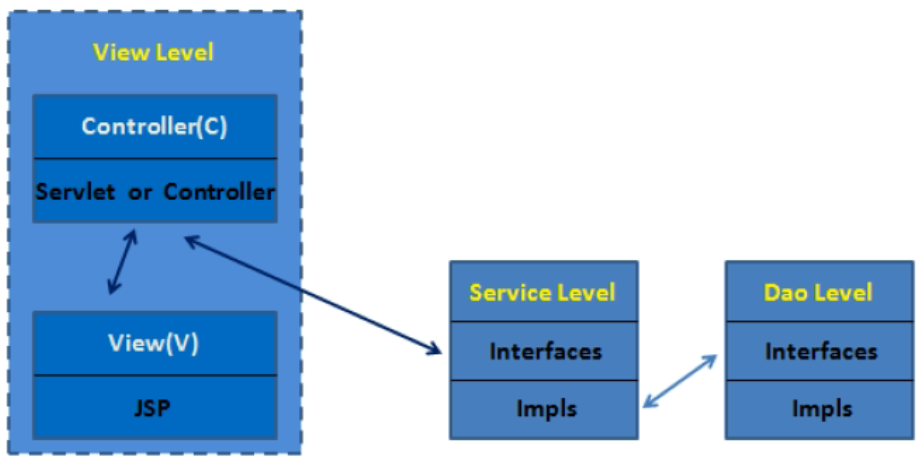
MVC

**三、关系**

**1、MVC与三层架构的关系**

MVC与三层架构很相似，但它们并不一样。如果以三层架构为背景，那么MVC的三个部分分别对应的是什么？

三层架构中的View层简单的说就是跟用户发生直接关系的层，MVC中的V和C就是这样的存在，所以MVC中的V和C均属于三层架构的View层。同时，我们知道MVC中的M（Model）包括了数据承载Bean和业务处理Bean，其中业务处理Bean分为Service或Dao对象，分别对应业务逻辑处理和数据库操作，相应的，它们对应的是三层架构中的Service层和Dao层。故，它们的关系如下图所示：



MVC与三层架构的关系

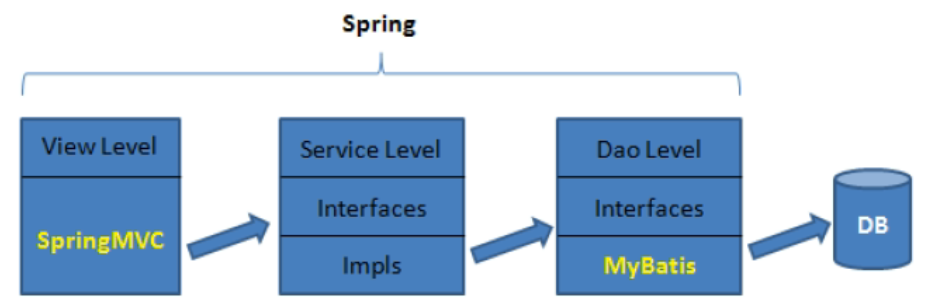
**2、SSM与三层架构的关系**

SSM即SpringMVC、Spring、Mybatis三个框架。它们在三层架构中所处的位置是不同的，即它们在三层架构中的功能各不相同，各司其职。

SpringMVC：作为View层的实现者，完成用户的请求接收功能。SpringMVC的Controller作为整个应用的控制器，完成用户请求的转发及对用户的响应。

MyBatis：作为 Dao层的实现者，完成对数据库的增、删、改、查功能。

Spring：以整个应用大管家的身份出现。整个应用中所有的Bean的生命周期行为，均由Spring来管理。即整个应用中所有对象的创建、初始化、销毁，及对象间关联关系的维护，均由Spring进行管理。



SSM与三层架构的关系