

七、MVC框架(Model-view-Controller)

7.1MVC概念

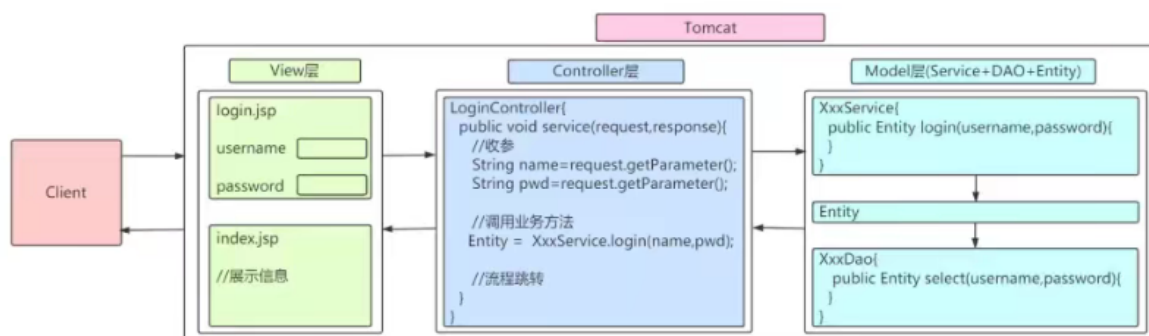
MVC又称为编程模式，是一种软件设计思想，将数据操作、页面展示、业务逻辑分为三个层级（模块），独立完成，相互调用

- 模型层（Model）
- 视图（View）
- 控制器（Controller）

7.2MVC模式详解

MVC并不是Java独有的，现在几乎所有的B/s的架构都采用了MVC模式。

- 视图View：视图即是用户看到并与之交互的界面，比如HTML（静态资源），JSP（动态资源）等等。
- 控制器Controller：控制器即是控制请求的处理逻辑，对请求进行处理，负责流程跳转(转发和重定向)。
- 模型Model：对客观世界的一种代表和模拟(业务模拟、对象模拟)。



7.3优点

- 低耦合性：模块与模块之间的关联性不强，不与某一种具体实现产生密不可分的关联性
- 高维护性：基于低耦合性，可做到不同层级的功能模块灵活更换、插拔
- 高重用性：相同的数据库操作，可以服务于不同的业务处理。将数据作为独立模块，提高重用性

7.4MVC在框架中应用

MVC模式被广泛用于Java的各种框架中，比如Struts2、SpringMVC等等都用到了这种思想。

7.5三层架构与MVC

7.5.1三层架构

View层 (表示|界面层)、Service层 (业务逻辑层)、DAO层(数据访问层)



7.5.2MVC与三层架构的区别

- MVC强调的是视图和业务代码的分离。严格的说MVC其实关注的是Web层。View就是单独的页面，如JSP、HTML等，不负责业务处理，只负责数据的展示。而数据封装到Model里，由Controller负责在V和M之间传递。MVC强调业务和视图分离。
- 三层架构是“数据访问层”、“业务逻辑层”、“表示层”，指的是代码之间的解耦，方便维护和复用。