SSM项目总结

本项目依然具有四大功能，增、删、改、查（逃

一、框架的力量

不过，我们终于用到了框架，即传统springMVC、spring和Mybatis。SpringFramework是一切的基础，DI的特性巧妙地化解了耦合，因为类被装进了spring容器；SpringMVC不仅比纯servlet更简单好用，而且更强大，只使用两个注解就能完成大多数功能；Mybatis在众多持久层框架中最终艰难脱颖而出，不仅是它简单而强大，而且还是因为解耦。包括Maven也让我们不用一个个拷包。

框架的用户是程序猿，事实证明，简单、强大、优秀的框架才能生存下来或者流行。从“用户”的观点看，框架大量使用注解和配置文件，而这两条“特点”恰好对应了简单强大、优秀，背后的思想就是封装重复操作、解耦。

二、SSM的难点

初学的难点就是不经常用的东西，这些就比较难记住，如果只配置一次就开始用，那就记不住怎么配置；如果用法也不常用，比如文件上传、那就记不住怎么编码，最后只能翻出老代码copy一下。

可见记住都是用出来的。怎么把不常用的用起来呢？我觉得就算老师的项目里没有明确要求，但自己一定要刻意练习，有些实在不常用的东西copy一下也罢，但是有些东西还是很重要的，就不要放过练习的机会，比如AOP事务。这点我自己在完成作业时也做得不好。

熟练使用是一项能力，也是下个难点的前提——源码实现、设计模式。这些内功依然是我们的薄弱点。

三、一点思考

编程套路是最容易掌握的，JavaEE的项目总结我甚至把从request到response的核心流程做了梳理，前一个项目最大的作用就是我们有了这个三层架构编程套路的基础，无非是servlet换成了spingMVC，queryRunner换成了MyBatis。作为初学者，我们最容易掌握的就是编程套路，也就是用的最多的那套东西：搭起三层架构，@RequestMapping对应url，封装请求参数，调用service层（本项目好像没有太复杂的业务逻辑），调用mapper层，在mapper.xml或者用注解书写sql语句，需要orm，落实到CRUD，返回。

框架本身是一种对原有东西的简化，本着程序猿偷懒的原则，能不重复造轮子绝不自己写，所以SSH没落了完全没毛病。我们就是要熟练掌握流行技术的用法，比如springboot，才能提升我们的开发效率，也适应时代。

另外依然一条漫长的道路就是学习源码中的编程思想，“变”的东西总在变，springboot全家桶敢说也肯定不是最终，但是“不变的”才是核心的竞争力。